

## Abstract

Washo is a Hokan language spoken in the area surrounding Lake Tahoe. In this paper, I will present an analysis of Washo plural reduplication using DIRECTED GRAPHS after Raimy (2000). The Washo example has proven a problem for other autosegmental approaches, which require a number of language-specific claims.

The analysis presented in this paper relies instead on a conception of a word as a directed graph. I claim that the brain can use an algorithmic assignment of precedence variables to segments easily selected using SIMPLIFIED BRACKETED GRIDS (SBG) to create such a graph. With only a few (likely lexical) exceptions, the SBG-derived template is sufficient to derive the plural for all forms containing a CV cluster without supplementary rules. I present an algorithm to handle these forms, and I argue that it is not surprising that a process which reduplicates a CV syllable should follow a similar, but not identical, algorithm to derive the plural of forms containing no CV cluster.

The success of a directed graph approach to reduplication relies on defining some predictable process of LINEARIZATION, a mechanism for turning linked segments into a single, unambiguous phonological word ready for phonemic interpretation. I also include a short discussion of the complexity of the linearization process.

The computation of reduplication presented for the Washo example in this paper accounts for the patterns found in that language in an algorithmic, computational tractable manner.

## 0 Introduction

The morphologies of languages of the world employ numerous devices to add meaning to a stem. Prefixing and suffixing are familiar to English speakers, as is word compounding. Less obviously, a change in stress can also add meaning to a word, frequently representing the difference between a noun and a verb as in the English verb and noun spelled *record*. Morphological processes vary far more greatly than is obvious from the English example. Instead of the prefixes and suffixes used in English, speakers of some language insert an affix into the word, making it an infix. Furthermore, when an English speaker uses an affix to modify the meaning of a word, it is always a specified affix, such

as *-s* to represent the plural, or *-ed* to represent the past tense. In some languages, however, an affix does not have any specific phonological information attached to it, and instead copies phonological information from another portion of the word. This process is called REDUPLICATION, as a part of the stem is duplicated in the final output.

Reduplication is extremely complicated, and has resisted the more simple analyses that linguists provide for other phonological and morphological processes. For that reason, many linguists have focused on finding a way to explain it. There are two common ways to view reduplication today. The first is that reduplication is simply an affixing process that obeys the rules of affixing in a language. Phonological information is copied to the phonologically empty affix. The other analysis uses optimality theory to require that reduplication occur and to specify some key aspects of the output form. I propose an analysis of reduplication based on DIRECTED GRAPHS, a computational tool often used by mathematicians and computer scientists.

In this paper, I will examine the reduplication pattern used to form the plural in the Washo language. The Washo (sometimes spelled Washoe) people live on the border between California and Nevada, surrounding Lake Tahoe. Although approximately 1,500 ethnic Washo are living today, of which around 300 still live in the Lake Tahoe area, the Washo language is highly endangered. Living speakers number around ten. Linguists debate the classification of the Washo language. Some place it in the Hokan language family along with several other indigenous languages of the United States and Mexico, but other scholars claim it is an isolate.

In this paper, I will discuss the phonology and morphology of reduplication. In §1, I present data introducing the reduplication pattern of Washo. Numerous linguists

have presented analyses of Washo plural reduplication, as it is complex enough to provide strong supporting evidence for the effectiveness of one analysis or shine light on the fatal flaw of another. I discuss some previous analyses in §2. In §3, I will provide an introduction to directed graphs, their computational benefits, and their applicability to phonology and morphology. I introduce my analysis of the Washo data using directed graphs in §4, and in §5 I offer some concluding remarks.

## 1 Reduplication in Washo

The plural in Washo is represented by reduplication. In general, the final CV syllable of a stem is reduplicated. The exact output form, however, is dependent on the structure of the stem. In this section, I will present several classes of Washo data.

### 1.1 Consonant-initial stems

Plural reduplication in consonant-initial forms is rather simple, but not trivial. I present three different types of consonant-initial forms. There do not appear to be any one syllable consonant-initial stems in Washo.

(1)	singular	plural	gloss
	'daʔa	da'ʔaʔa	mother's brother
	'gewe	ge'wewe	coyote
	'guʔu	gu'ʔuʔu	mother's mother
	hañakmuwe	hañkmuwewe	elks

To form the plural of forms ending in VCV such as those above (ignoring, for now, stress), the final syllable of the stem is reduplicated. The pattern is slightly different for forms ending in VCVC.

(2)	singular	plural	gloss
	'sukuʔ	su'kukuʔ	dog
	'damal	da'mamal	to hear
	'p'isew	p'ise sew	ear
	baloxat	baloxaxat	bows
	malosañ	malosasañ	stars

The forms in (2) demonstrate that the reduplicant is simply the final CV of the stem, not the final syllable, since the codas of the final syllables in the words in (2) are not reduplicated. Forms ending in CCV(C) complicate the pattern further.

(3)	singular	plural	gloss
	'ʔewš̥iʔ	ʔe'š̥iwš̥iʔ	father's brothers
	net'us	net'unt'us-u	old women (with suffix)
	'saksag	sa'saksag	father's father's brother
	'mokgo	mo'gokgo	shoe

By observing the words in (3), it is clear that a consonant cluster interferes with the basic pattern. The reduplicant, still the final CV cluster of the stem, appears to “jump” over the coda of the preceding syllable to surface before it.

### 1.2 Vowel-initial stems

Consonant-initial forms in Washo demonstrate the basic reduplicative pattern of the language. Vowel-initial forms, however, introduce a further twist. In these forms, as seen in (4), the first vowel appears to be deleted.

(4)	singular	plural	gloss
	'ahad	'hahad (*a'hahad)	across <sup>1</sup>
	'aŋkaš	'kaŋkaš (*a'kaŋkaš)	hollow

All of these multi-syllabic, vowel initial forms support the argument that the only difference between consonant and vowel-initial forms is that the first sound is deleted from the vowel-initial forms. Consider, however, single-syllable forms. It appears that all one-syllable forms in Washo begin with vowels.

(5)	singular	plural	gloss
a.	'akd	'kakd	slowly
	'awd	'wawd	over the summit
	'aʔm	'ʔaʔm	to the west, from the east
	'a:m	'ma:m	to hit with body part
b.	'ips	'peps	up from the surface
	'išm	'šėšm	to sing, song
	'ilm	'lelm	under, underneath
	'im	'mem	out from
	'iw	'wew	in a certain direction

---

<sup>1</sup> Although I am far from an expert on the structure of Washo, I suspect that the language requires agreement among nouns, verbs, adverbs, adjectives and prepositions, explaining why such words have plural forms.

The words in (5) a and b appear to reduplicate in a similar manner. The first consonant is reduplicated before the first vowel. The forms in b differ from those in a because a vowel change occurs in addition to reduplication of the first consonant. This vowel change will be discussed later.

### 1.3 Exceptional forms

Washo also contains a number of forms that appear to exhibit an irregular reduplication pattern. The stems fall into the classes described above, but the behavior of these words is rather surprising. I present them here because they will aid the reader in careful consideration of the data, but I withhold a complete discussion and analysis until a framework for consideration of Washo plural reduplication is established.

(7)	singular	plural	gloss
a.	First vowel changes from i to e, second vowel deleted		
	'iʔiʃ	'ʔeʔʃ (*'ʔiʔiʃ)	(empty stem)
	'iʔib	'ʔeʔb (*'ʔiʔib)	cry, weep
b.	Pair of consonants reduplicates		
	'alʔmul	'ʔmolʔmol	big and round
	'iʔiw	'ʔweʔw (*'ʔiʔiw)	to eat
c.	Second y is deleted		
	'ayam	'ya:m (*yayam)	to hit with an instrument
	'ayaw	'ya:w (*yayaw)	black

'ayuk	'yo:k	(*yoyok)	parent in law
-------	-------	----------	---------------

d. First consonant of cluster reduplicates

'helme	he'lhelme	(*he'melme)	three
--------	-----------	-------------	-------

'hesge	he'sesge	(*he'gesge)	two
--------	----------	-------------	-----

'ašdim	'šašdim	(*'dišdim)	to hide
--------	---------	------------	---------

'mayŋa	ma'yayŋa	(*ma'ŋayŋa)	fawn
--------	----------	-------------	------

'aʔšam	'ʔaʔšam	(*'šaʔšam)	to lie
--------	---------	------------	--------

maʔšaʔ	maʔaʔšaʔ	(*mašaʔšaʔ)	brother's child
--------	----------	-------------	-----------------

e. no logical explanation

emlu	'mumlu	(*'lumlu)	to eat, food
------	--------	-----------	--------------

I provide an analysis or explanation of each of these classes in §4.

## 2 Previous analyses

A number of phonologists have attempted to classify Washo plural reduplication. Since the pattern is complicated and difficult to analyze, it has been used as evidence for and against general approaches to reduplication numerous times. The most recent previous analyses fall into two categories: those using a modified Marantzian approach, and those using optimality theory. The discussion of reduplication presented in Jacobsen's 1964 grammar, and the review of Jacobsen's work by Winter (1970) address reduplication in a

pre-Marantzian approach that is more of a description than an analysis and is so complex and imprecise that I have decided against presenting it.

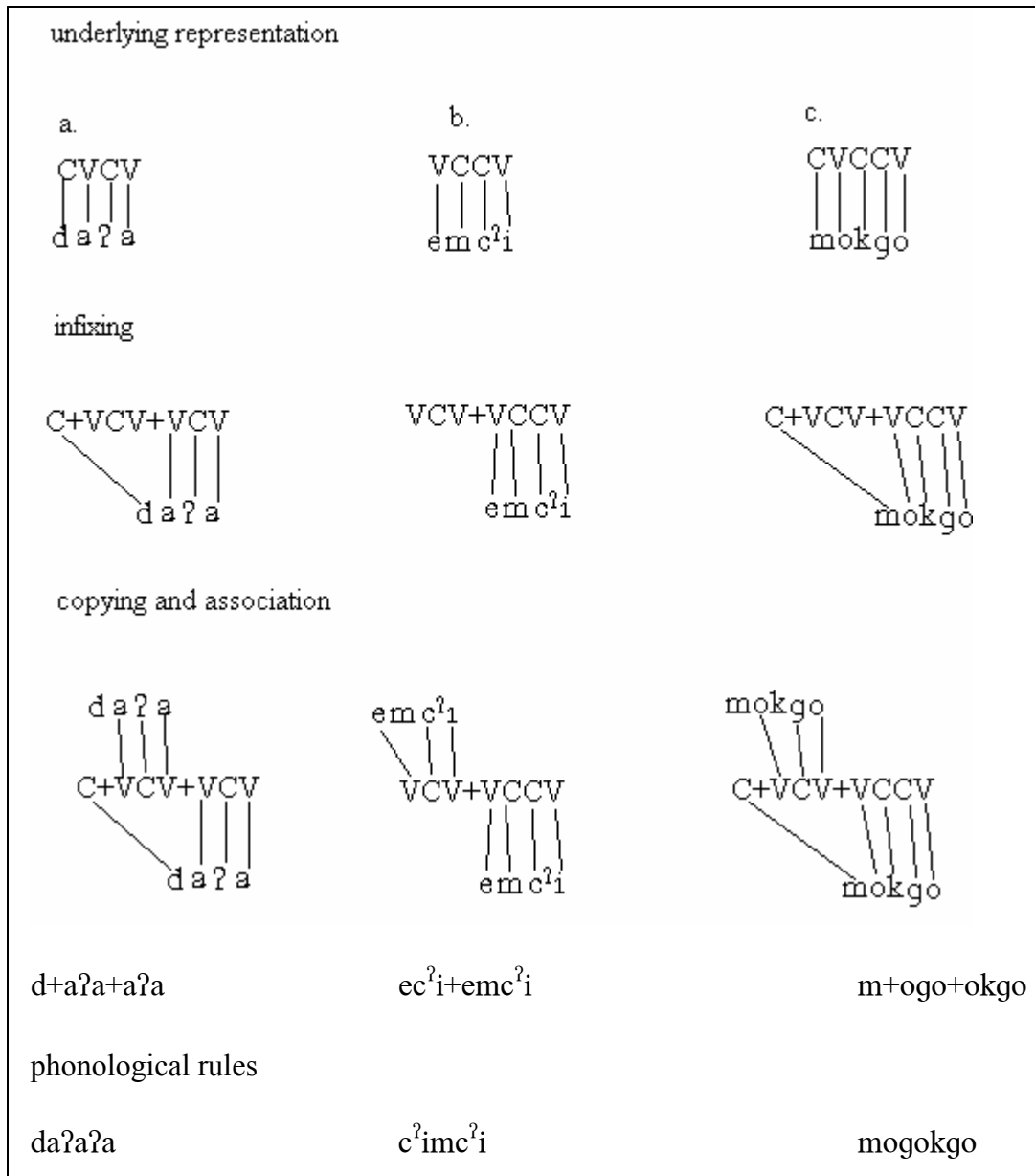
### **2.1 Marantzian analyses (Broselow and McCarthy)**

Alec Marantz (1982) proposed that reduplication is an unspecified type of affixation. Instead of affixes carrying phonological information, Marantzian analyses are based on appending morphological information onto the CV skeleton of a word. Phonological information is associated with the empty CV slots by copying and reassociating. Marantz's general idea inspired a novel approach to reduplication. Linguists have since added more specific rules and stipulations, but the analysis in this section makes use of Marantz's primary insight about unspecified affixes.

Broselow and McCarthy (1983) extend Marantz's general model to allow for further specification. They claim that a purely Marantzian model fails to account for complicated instances of internal reduplication such as Washo because it does not supply necessary constraints for association of phonetic material to unspecified segments. In order to account for Washo plural reduplication, Broselow and McCarthy propose the addition of an unspecified VCV affix in the environment  $\#(C)\_$ . The stem is then copied, and association proceeds from right to left. Consider the derivations of plural forms according to Broselow and McCarthy in (8).

(8)



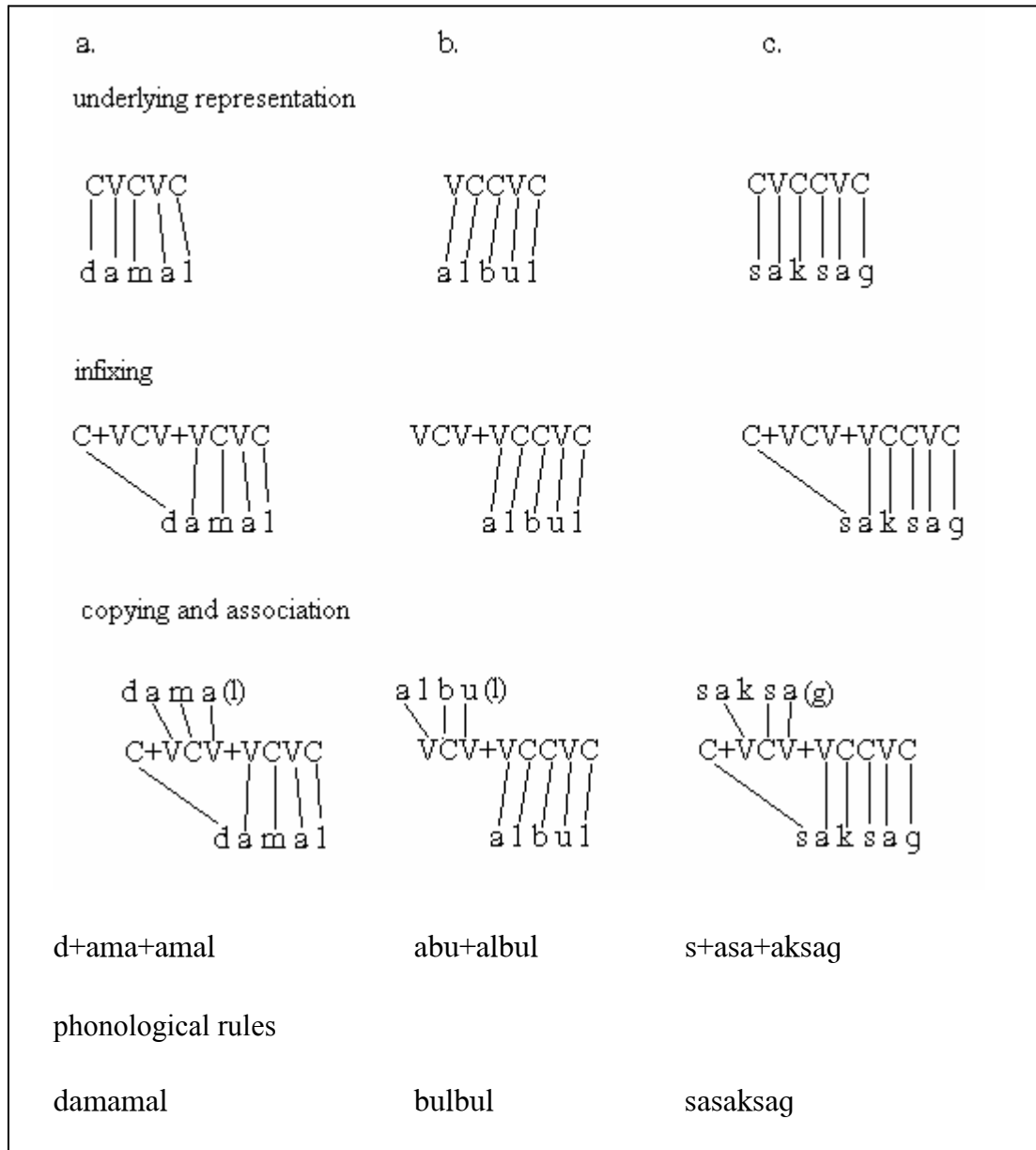


Note that not every segment of the copied stem must be associated.

The examples in (8) demonstrate the need for a set of vowel coalescence rules in order for this analysis to account for the data. Broselow and McCarthy propose a complicated but consistent set.

Broselow and McCarthy have to make a further stipulation in order to analyze Washo plural reduplication. They require that the final consonant of a word be extrametrical. Consider the examples in (9).

(9)



Note that in addition to vowel coalescence rules and extrametricality, Broselow and McCarthy assume that an unstressed vowel may not occur in word initial position. Stress

in Washo is always on the penultimate syllable, and stress is calculated after reduplication. More information about stress is presented in §4.4.1. This vowel deletion rule accounts for the forms in (8)b and (9)b.

Considering reduplication to be an unspecified form of affixation simplifies a phonological and morphological process that has been notoriously difficult to analyze to one of the most basic types of concatenative morphology. It is a seductive way to analyze reduplication. Although Broselow and McCarthy claim to analyze the majority of the Washo data, they provide no analysis for the single syllable stems presented by Yu (2003). Furthermore, they rely on several language-specific assumptions, including extrametricality, vowel coalescence, and vowel deletion rules.

## **2.2 Optimality Theory (Yu)**

Early generative, or rule based phonology, depends on making a list of rules for the formation of words. The sequence of these rules must be specified in order to produce the correct output. Frustrated by apparent paradoxes of rule ordering, some phonologists began to develop OPTIMALITY THEORY, which depends on violable constraints instead of hard and fast rules. Optimality theory (OT) views the brain as an accepting machine instead of a generating machine. It compares potential output forms of a word, finding the one that violates the fewest or the least relevant constraints. The “winning” form is the final output.

OT analyses make up much of modern phonology. Alan Yu (2003) uses OT to analyze Washo plural reduplication. In the framework of OT, Yu claims that the reduplicant in Washo is the final, monomoraic segment and that its surface location is dictated by the stress pattern in the language. He provides an analysis of the stress pattern

in Washo using constraints about footing and alignment, and he claims that the reduplicant is aligned with the stressed foot, in the same position as stress. To account for the placement of the reduplicant in a word with an internal consonant cluster before the final vowel, Yu invokes syllable weight as a factor in footing. He adds several further constraints tying the number of output syllables or moras to the input number. In (10) is an example of an OT chart predicting the output form of the plural of *saksag*.

(10) From Yu (2003:161)

/sa <sup>μ</sup> k.sa <sup>μ</sup> g, RED <sup>μ</sup> /	RED	MAX <sub>μ</sub>	WSP	WBP	PARSE	DEP <sub>μ</sub>
a. sa <sup>μ</sup> .(ˈsɑ <sup>μ</sup> k <sup>μ</sup> .sa <sup>μ</sup> g)				*	*	*
b. sa <sup>μ</sup> k.(ˈsɑ <sup>μ</sup> .sa <sup>μ</sup> g)				*!*	*	
c. sa <sup>μ</sup> k <sup>μ</sup> .(ˈsɑ <sup>μ</sup> .sa <sup>μ</sup> g)			*!	*	*	*
d. sa <sup>μ</sup> k.(ˈsa <sup>μ</sup> g <sup>μ</sup> .sa <sup>μ</sup> g)	*!			*	*	**
e. sa <sup>μ</sup> k.(ˈsa <sup>μ</sup> g.sa <sup>μ</sup> g)				*!*	*	
f. sa <sup>μ</sup> .(ˈsɑ <sup>μ</sup> k <sup>μ</sup> .sa <sup>μ</sup> g <sup>μ</sup> )			*!	*	*	*
g. sa <sup>μ</sup> k.sa <sup>μ</sup> .(ˈsɑ <sup>μ</sup> g <sup>μ</sup> )				*	*!*	*

Note that  $\mu$  represents a mora break. The constraints are as follows:

RED: Reduplicant must be one mora

MAX<sub>μ</sub>: Number of moras in output corresponds to the number of moras in the input

WSP: A heavy syllable must be stressed

WBP: A coda consonant is moraic

PARSE: Every syllable is footed

DEP<sub>μ</sub>: Number of moras in input corresponds to number of moras in output

\* = violation, gray square = fatal violations, pointing finger = correct form.

Syllable weight does not appear to be a factor in any other calculation in the language, and it does not change the position of stress when it occurs word-finally. It therefore seems surprising that syllable or moraic weight would determine the placement of the reduplicant. As I am no expert on the structure or evolution of Washo and related

languages, I will not foray into a discussion of the advisability of use of syllable weight as a factor in reduplicant placement. My only further criticism of Yu's analysis of Washo is that his attempts to fit a single analysis to all forms may make accounting for exceptions more difficult, as noted in § 4.3.

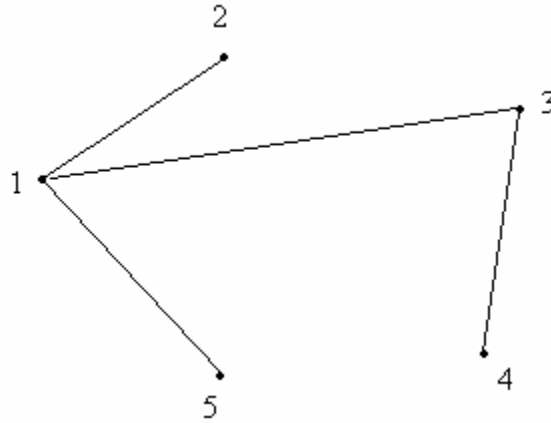
### **3 Directed graph analysis**

In order to discuss phonological computation in any depth, it is necessary to select a model of data storage upon which to perform computation. In this section, I define DIRECTED GRAPHS mathematically, compare them to several other data structures and discuss some conventions in representing words as directed graphs.

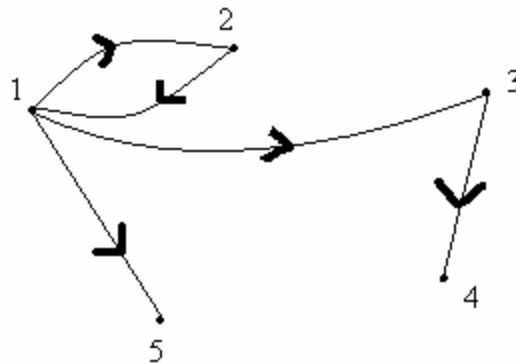
#### **3.1 About directed graphs**

According to Discrete Mathematics with Graph Theory a graph is “pair  $(V, E)$  of sets,  $V$  nonempty, and each element of  $E$  a set of two distinct elements of  $V$ ” (Goodaire and Parmenter, 2002:286). The elements of  $V$  are generally known as VERTICES, and the pairs in  $E$  are called EDGES. The vertices of a graph are generally represented by dots, while the edges are represented by lines drawn between the dots. A directed graph is a graph where the order within a pair in  $E$  matters. The elements of  $E$  in a directed graph are called arcs, and are represented by arrows.

(11) a      The graph with  $V=\{1, 2, 3, 4, 5\}$  and  $E=\{\{1, 2\}, \{1, 3\} \{1, 5\} \{3, 4\}\}$



- b The directed graph with  $V=\{1, 2, 3, 4, 5\}$  and  $E=\{(1, 2), (1, 3), (1, 5), (2, 1), (3, 4)\}$ <sup>2</sup>



A visual example of the difference between a graph and a directed graph is apparent in (11). The arc set of the directed graph in b contains the elements  $(2, 1)$  and  $(1, 2)$ . Since the order of vertices in an arc pair matters, these are different arcs, as is evident from the picture. Putting both  $\{1, 2\}$  and  $\{2, 1\}$  in the edge set of a graph, however, would be redundant. Since the order of vertices in an edge pair does not matter, these edges are identical.

### 3.2 Why directed graphs?

<sup>2</sup> It is a mathematical convention to use parentheses,  $( )$ , to represent an ordered pair or set, but curly brackets,  $\{ \}$ , to represent an unordered pair or set.

Raimy (2000) proposes using directed graphs to represent phonological words. He gives a detailed explanation of predictions facilitated by directed graphs. I will briefly summarize some of the benefits of directed graphs. Raimy (2000) provides a more detailed explanation.

It is clear that graphs are not an adequate representation of a phonological word. Since graphs do not specify direction, any edge can be traversed in any direction. Representation of words as graphs would prevent any meaningful discussion of precedence. Many phonological processes depend on precedence and linearity. A phonological calculation might only occur in a beginning-to-end manner. A graph has no way of indicating the beginning of the word, and therefore the order of the segments. In the graph in (12), the segments could be combined in the order 1234, or 4321, or 12321, although not 14 or 132, since the edges have no specified direction of traversal.

(12)



Since graphs are incapable of capturing the importance of specified order, they are far from sufficiently restrictive to provide a basis for the phonological word.

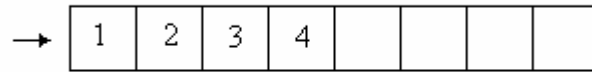
One possible representation of a phonological word is the array, as seen in (13)a. An array is a set of spaces with a specified beginning. Arrays clearly indicate the beginning of a word, and make suffixing particularly easy, since the morphological information is simply appended to the end of the word as in (13)b. Other types of



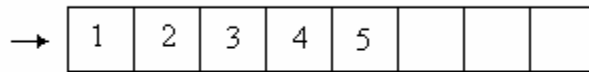
affixing, however, are quite complicated using an array. In order to add a prefix, in (13)c, the phonological information present in the stem must either be copied further down the array, or the prefix, followed by the stem, must be copied into a new array, since the beginning of an array is specified, so no more segments can be added before the beginning. Infixing, in (13)d, presents a similar problem.

(13)

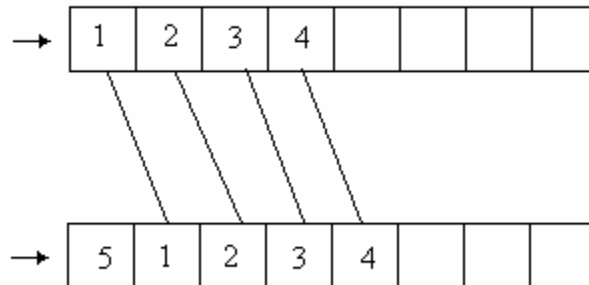
a.



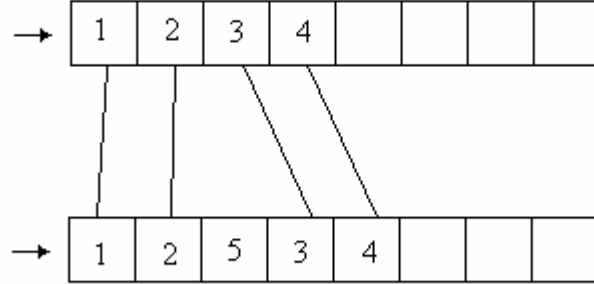
b. suffixing



c. prefixing



d. infixing



Prefixing and suffixing are both extremely common processes in languages. A phonological model such as the array, in which the processes for different types of affixing require such large variation in required computation, would predict that the process requiring less computation would occur with great frequency, while the process requiring more computation would be relatively rare. Prefixing and suffixing are both extremely common morphological processes, so it would be surprising to find a computational model in which one is far more complicated than the other.

Directed graphs are a phonological model that combines the most useful elements of arrays and graphs. Graphs allow for easy addition and deletion of phonological units by adding an element to the edge set, while arrays give a strict precedence structure by specifying the order of segments. Directed graphs specify order, but also allow for relatively easy modification.

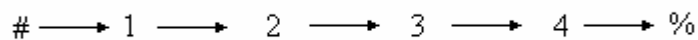
It is worth noting that an array is a type of DATA STRUCTURE, while graphs and directed graphs are ABSTRACT DATA TYPE. Data structures are ways of sorting data that include predefined notions of operations. Note that in the discussion of arrays, I presented a method of copying and computing. In an abstract data type, however, data is stored but no operations such as copying or linking are specified. Choosing directed graphs, an

abstract data type, to represent phonological units means we need to do a little bit more work to define the operations we will need. Directed graphs do not have a built-in method for combination or for compression into a single, linear unit. It would be convenient to be able to take advantage of these predefined notions by using a data structure to represent a phonological word, but we have evidence to show that an array, the most likely usable data structure, is not sufficient. I therefore devote §4.5 to a discussion of linearization, a proposed operation for turning a directed graph into a word whose segments occur in a defined order.

### 3.3 Conventions of Directed Graphs

Following Raimy (2000), I will represent directed graphs as pictures,<sup>3</sup> as in (14).

(14)



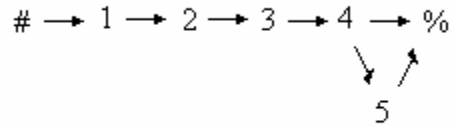
The information in (14) will generally be taken to represent a stem. The symbols # and % are not actually part of the word; they represent the ideas of “coming from nothing” and “going to nothing,” respectively. The arrows are arcs, and the numbers represent phonological segments. I will also call the elements of E, the arc set, PRECEDENCE VARIABLES. As with a general directed graph, a phonological directed graph, or any computation upon it, can be represented by either its picture or its precedence variable set. Using the above conventions, it is relatively simple to represent extremely varied phonological phenomena such as prefixing in (15) a, suffixing in (15) b, infixing in (15) c, and reduplication in (15) d.

---

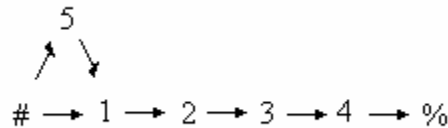
<sup>3</sup> In strict mathematical terms, a graph is the pair (V, E), not a picture. A picture is a visual representation of a graph.

(15)

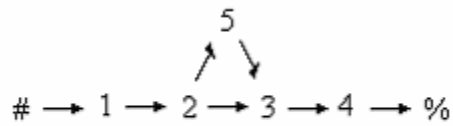
a.



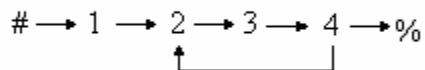
b.



c.



d.



Although these graphs contain all of the arcs necessary to traverse the desired path, they also contain additional arcs that will not be traversed. In order to remove any nondeterminism, or uncertainty, from the graph, a process called linearization turns the word into a string of segments. There is much debate as to the specifics of linearization. I

will discuss some theories in section §4.5 after presenting an analysis of Washo plural reduplication.

#### 4.0 Analysis of Washo reduplication

In this section, I will present a directed graph analysis of plural reduplication in Washo. I hope to demonstrate that directed graphs are not only a simple tool for representing phonological words that facilitates computation, but also that a directed graph analysis accounts for the data in this complicated language at least as successfully as any published analysis to date.

##### 4.1.1 Consonant-Initial Stems

Consonant and vowel-initial stems trigger slightly different reduplication patterns in Washo. In this section, I present an analysis of consonant-initial forms.

(16)	singular	plural	gloss
	'daʔa	da'ʔaʔa	mother's brother
	'gewe	ge'wewe	coyote
	'guʔu	gu'ʔuʔu	mother's mother's
	hañakmuwe	hañkmuwewe	elks

The data in (16) exhibit a simple and predictable final syllable reduplication pattern.

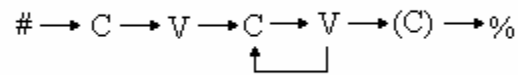
From the data in (17), however, it is clear that the reduplicant is only the first consonant and vowel of the final syllable.

(17)	singular	plural	gloss
	'sukuʔ	su'kukuʔ	dog
	'damal	da'mamal	to hear

'p'isew	p'ise sew	ear
baloxat	baloxaxat	bows
malosañ	malosasañ	stars

Given the data seen thus far, the following appears to be a likely directed graph representation for Washo stems.

(18)



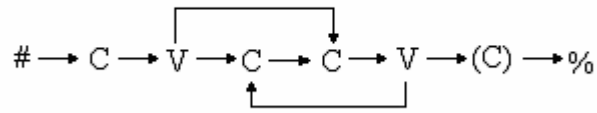
It is clear from examination of further data, however, that the precedence relation represented by the directed graph in (18) is incomplete.

(19)	singular	plural	gloss
	'ʔewšičʔ	ʔe'šiwšičʔ	father's brothers
	net'us	net'unt'us-u	old women (with suffix)
	'saksag	sa'saksag	father's father's brother
	'mokgo	mo'gokgo	shoe

In the examples given in (19) the reduplicant precedes the coda of the previous syllable.

These data suggest a different directed graph structure for words with penultimate syllable codas.

(20)



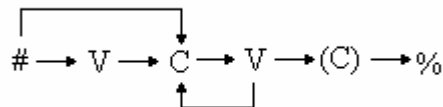
#### 4.1.2 Vowel-Initial Stems

Washo treats stems beginning with a vowel somewhat differently. It appears that the first vowel of these vowel-initial forms is not articulated. I will argue in §4.2, however, that the vowel is skipped by an arc in the directed graph rather than deleted.

(21)	singular	plural	gloss
	'ahad	'hahad	across
	'aŋkaš	'kaŋkaš	hollow
	'ašun	'šošoŋ	red <sup>4</sup>

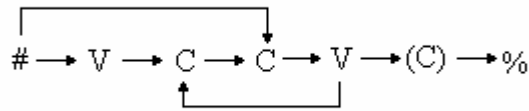
Vowel-initial words follow almost the same precedence relation as consonant-initial forms. Additionally, they have a mechanism for skipping the vowel. In (22) I propose directed graphs for vowel initial forms, depending on the presence or absence of a first syllable coda.

(22)



or

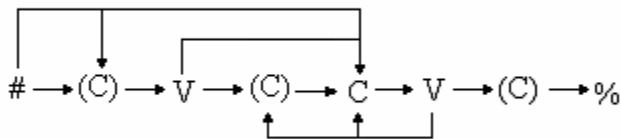
<sup>4</sup> Yu (2003) and Urbanczyk (1993) give convincing evidence that [o] occurs only in a stressed syllable or in the presence of another stressed /o/. The underlying form of this word is therefore /ašon/.



### 4.1.3 General Analysis of Regular Forms

As is apparent in §4.1.1 and §4.1.2, all Washo forms containing a syllable with an onset can be represented with similar directed graphs. It is, in fact, easy to modify the directed graphs shown in this section so that all forms with at least one CV syllable follow the same one which I present in (23). Although some of these arcs may be vacuous, I claim that a language with a single prototype with vacuous arcs is simpler than a language with many prototypes and no vacuous arcs. This directed graph looks much more complicated due to the possible differences among two syllable words.

(23)

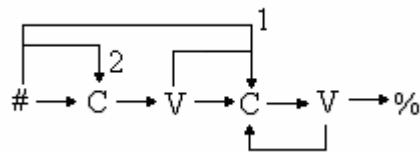


Note the presence of several arcs with multiple possible destinations in (23). If two arcs begin at the same place, I claim that the ELSEWHERE CONDITION (Kiparsky, 1973) requires that an arc to an optional consonant be followed if one is available. Imagine a situation in generative phonology in which two similar rules appear to apply. The notation of rule 2 is a subset of the notation of rule 1. The elsewhere condition states that whenever rule 1, which is more specific, can apply it will and it will furthermore block

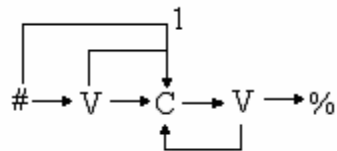


rule 2 from applying. Phonologists have found this condition to be particularly useful in accounting for the interaction of multiple rules, especially when dealing with irregular forms. In the case of directed graphs, an arc going to an optional consonant is more specific than an arc going to a mandatory one. If both exist, in (24) a, then both arcs will be present. If only the mandatory consonant exists, as in (24) b, a single arc will be present, so the arc to the mandatory consonant is a subset of the notation of the arc to the optional one. By the elsewhere condition, the arc to the optional consonant precludes the arc to the mandatory one. In example (24) a, only arc 2 will surface, while in (24)b it is clear that only arc 1 will surface in b since there is no other relevant arc.

(24) a.



b.



It is important to note that consonants within a given stem are not mandatory or optional; they are required to form that stem. This discussion refers to the method by which this model is fit to actual words in the language. An optional consonant is one that is not specified by the definition of the form we are considering: a two-syllable word including a CV cluster.

As discussed in §3, a directed graph can be represented by a set of elements, its vertices, and a set of ordered pairs, its arcs, or precedence variables (Raimy, 2005). Although a list of precedence variables is less visual than the above directed graph, it is slightly more precise in that it easily distinguishes onsets from other consonants. We cannot specify whether consonants are onsets as easily in (23), as some arcs are specified to go to a consonant, regardless of position, while others are specified to go to onsets. It is easy to make these specific invocations in the list of precedence variables in (25). The precedence variable set does not, therefore, require the elsewhere condition to adapt to specific forms.

(25)

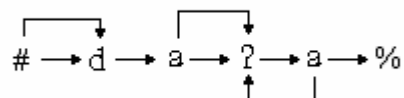
{#}, {first onset}  
 {penultimate vowel}, {last onset}  
 {last vowel}, {penultimate vowel \_}

These precedence variable pairs are simply another representation of the graph in (23). It is interesting to note that the result of the directed graph (where no onset/coda distinction is made) is equivalent to the precedence variables in (25) where onsets, not general consonants are calculated when the elsewhere condition is used to limit its application.

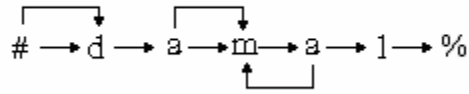
#### 4.1.4 Examples

Using the analysis given above, it is easy to explain plural reduplication patterns seen in Washo.

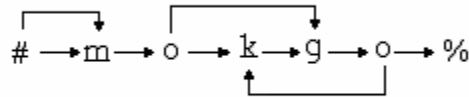
(26) a.



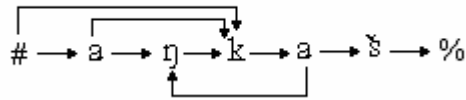
b.



c.



d.



The directed graphs in (26) are examples of how the general directed graph in (23) applies to the classes of data seen thus far. Examples (26) a and b consonant-initial forms with no penultimate coda, (23) c shows how the analysis accounts for the skipping of a coda in a consonant cluster, and (26) d is an example of a vowel initial form. These specific cases of the general directed graph above predict the correct plural forms of the given words.

The directed graphs represented above are not yet words as a speaker would utter them. A speaker cannot phonetically represent one of the arrows in the above diagrams. In order for these models to become the phonological utterances that a speaker can produce and a listener can recognize, they must be subjected to some process to predict the correct path through the graph, remove the arrows, and produce a sequence of segments which speakers would recognize as a phonological word. This process, linearization, will be discussed in §4.5.

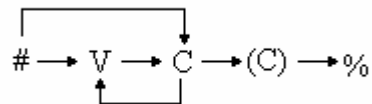
## 4.2 Forms without Onsets

As discussed in §4.1.3, it is possible to use a single set of precedence variables to account for all forms containing a syllable with an onset. Several of the arcs in this model of reduplication are anchored on syllable onsets. Given the model in (23) and (25), it is unclear what prediction would be made with regards to plural reduplication in words lacking an onset.

(27)	singular	plural	gloss
	'akd	'kakd	slowly
	'awd	'wawd	over the summit
	'aʔm	'ʔaʔm	to the west, from the east
	'a:m	'ma:m	to hit with body part

None of these stems contain an onset, so the plural reduplication pattern that they take is not surprisingly different from the one described in §4.1, which relies heavily on calculation of syllable onsets. These forms can be calculated by a directed graph shown in (28) or a set of precedence variables in (29).

(28)



(29) {#}, {first vowel\_}

{last vowel\_}, {last vowel}<sup>5</sup>

Note that the relationships described in (28) and (29) preserve much of the structure of the precedence relations stems with onsets. In each case, the reduplicant contains an onset in the plural form, although in the case of forms without onsets the onset in the plural form was originally a coda consonant. Furthermore, all of these forms begin with vowels. It is not surprising to find that, similarly to the vowel initial forms in (21), the initial vowels in forms without onsets are skipped. Interestingly, these onsetless forms provide a piece of evidence that vowels are in fact skipped in the precedence relationship, not merely deleted. In these forms, the vowel appears at a later point in the word, so it cannot have been deleted. It makes sense to assume that vowels are uniformly skipped, not deleted, in Washo plural reduplication.

Several other types of stems act similarly to the forms shown in (27). Most can be explained by determining the underlying form of a stem.

(30)	singular	plural	gloss
	'ips	'peps	up from the surface
	'išm	'šešm	to sing, song
	'ilm	'lelm	under, underneath
	'im	'mem	out from
	'iw	'wew	in a certain direction

---

<sup>5</sup> Washo appears to allow onsetless syllables only at the beginning of a word, so the only words without onsets must have only one syllable. It is therefore difficult to be certain that the variable referenced is the last, not the first. I will assume “first vowel” and “last vowel” in places where they seem to correspond most easily to the model in the previous section.

Following Urbanczyk (1993), I will assume that Washo prohibits [e] in onsetless syllables<sup>6</sup>. These words are repaired in the singular by raising the /e/ to [i]. Once the underlying forms of these stems are considered, the precedence relation shown in (30) and (29) explains their plural forms.

(31)	singular	plural	gloss
	'iʔiʃ	'ʔeʔʃ (*'ʔiʔiʃ)	(empty stem)
	'iʔib	'ʔeʔb (*'ʔiʔib)	cry, weep

Yu (2003) notes that the underlying forms of words such as these in the form VCVC where the first C is a glottal stop can be seen by looking at affixed forms.

(32)	iʔib	to cry	ʔ-iʔb-i	she's crying
	iʔiw	to eat something	k-iʔw-i	he's eating it

By assuming underlying stems with no second vowel, these data can be treated similarly to those in (27).

### 4.3 Exceptions

Several plural forms of Washo appear to be exceptions to the reduplication pattern described above. I will present the classes of exceptions and discuss why they are surprising below.

(33)	singular	plural	gloss
------	----------	--------	-------

---

<sup>6</sup> Note that Yu modifies this claim, arguing that it only applies to single-syllable words and citing the word *elmu* (to eat, food) as an example. This word is the only example he gives, and it also proves to be an unexplainable example of plural reduplication in Yu's analysis as well as mine. I therefore hesitate to form any conclusions based solely on this example.

'alʔmul	'ʔmolʔmol <sup>7</sup>	big and round
'iʔiw	'ʔweʔw <sup>8</sup> (*ʔeʔw)	to eat

The forms in (33) are surprising because they are the only forms in Washo that appear to reduplicate two consonants. It is interesting to note that in both forms the first reduplicated consonant is a glottal stop. Also, *big and round* is one of a very few examples of a cluster of three consonants in the language. The fact that these forms are unusual seems to suggest that glottal stops are different from other consonants. I suspect that there is some difference in the language between the full glottal stop and glottalized consonants, which either sound the same or were transcribed identically. If the glottal stops and the following consonants are considered to be a single sound in the forms in (33), their plural forms are completely regular.

(34)	'ayam	'ya:m	(*yayam)	to hit with an instrument
	'ayaw	'ya:w	(*yayaw)	black
	'ayuk	'yo:k	(*yoyok)	parent in law

The forms in (34) contain a [y] between two vowels. Yu proposes a ban on intervocalic y when the two vowels are the same and the first is stressed. He claims that this ban applies only to derived forms, so the stems in (34) that violate his rule are in fact acceptable. I propose either a more restricted rule, where such a ban only occurs after another y, or some mechanism that causes the vowel alone to be the reduplicant, excluding the syllable onset. I leave the y question open for further research.

---

<sup>7</sup> It is unclear what form should be expected here as there is no other example of three sequential consonants in Washo.

(35)	'helme	he'lelme	(*he'melme)	three
	'hesge	he'sesge	(*he'gesge)	two
	'ašdim	'šašdim	(*dišdim)	to hide
	'mayŋa	ma'yayŋa	(*ma'ŋayŋa)	fawn
	'aʔšam	'ʔaʔšam	(*šaʔšam)	to lie
	maʔšaʔ	maʔaʔšaʔ	(*mašaʔšaʔ)	brother's child

The irregular forms in (35) appear to be quite similar to the forms without an onset discussed in §4.2. I therefore propose that these forms are a type of lexical compound. I claim that the reduplicative process involves only the first vowel and the following consonants, which is probably represented by a lexical bracket inserted in the simplified bracketed grid for these words. Jacobsen also finds these forms to be surprising, and he notes that these exceptional forms “may indicate that there were (or are still) morpheme boundaries immediately after the consonant clusters, which would make the stems regular monosyllabic ones” (1964:329), a claim that would be consistent with lexically added bracket in the metrical structure of the word.

(36)	emlu	'mumlu	(*'lumlu)	to eat, food
------	------	--------	-----------	--------------

The exceptional form in (36) defies all attempts at explanation. The stem is especially unusual in that it begins with an [e], deemed in §4.2 to be highly marked in Washo. I am forced to conclude that this form is a lexical exception.

#### 4.4 Other Prosodic Calculations



Reduplication in Washo affects other prosodic factors in the language, such as stress, discussed in §4.1, and vowel length, discussed in §4.2.

#### 4.4.1 Stress

Stress in Washo is extremely predictable, falling always on the penultimate syllable.

(37)	a.	singular	plural	gloss
		'guʔu	gu'ʔuʔu	mother's mother's
		'daʔa	da'ʔaʔa	mother's brother
		'gewe	ge'wewe	coyote
	b.	'ahad	'hahad	across
		'ʔeʔel	ʔe'lelel	mother's father
		'ʔewšič	ʔe'šiwšič	father's brothers
		'aŋkaš	'kaŋkaš	hollow
	c.	le-'ʔeʔel-iʔ	my daughter's child (of a man)	
		le-'guʔu-iʔ	my daughter's child (of a woman)	
	d.	singular	plural	gloss
		'sesm	se'sesm	to vomit

From the data in (37) a, it is clear that stress is calculated after reduplication. Stress falls on a new syllable in the plural when reduplication affects the penultimate syllable. The data in (37) c, however, show that affixing occurs after the stress calculation. The only exception to the penultimate stress pattern occurs in (37) d. It is interesting to note that the word *sesm* is the only one in Yu's data set that contains both an onset and branching

in the coda. One way to predict this result is to claim that Washo stress is sensitive to ultra-heavy syllables, which I will define as syllables with branching in the coda. There is no evidence, however, that Washo is sensitive to syllable weight. As is evident from the data in (37) b, the stress calculation makes no use of heavy syllables. If it did, one would expect the forms in part b to exhibit final stress, since in each form the final syllable is heavy. Instead I propose that the plural of *sesm* retains stress on the final syllable for historical reasons, either because it is a loan word from a language with a different stress pattern or because the word originally contained a second vowel, between the s and the m. When the vowel was lost, the reduplication pattern adapted to the new form but the memorized stress from the original form was maintained. The extremely regular stress pattern of Washo presents no problem for a directed graph analysis of plural reduplication.

#### 4.4.2 Vowel Length

Washo exhibits an extremely unusual pattern of vowel length shift in reduplicated forms.

(38)	singular	plural	gloss
	mem'de:wi	memde'wi:wi	deer
	'me:hu	me'hu:hu	to be a boy
	'wa:šiw	wa'š'i:šiw	Washo

Either Washo has some mechanism for spreading of vowel length, or both vowels in the word are marked for length, which only surfaces in stressed syllables. I will assume a variation of the latter. Harrison and Raimy (2004) counter the popular assumption that the least phonologically marked element, here the short vowel, is stored by the speaker. They argue instead that the most marked reasonable form is stored. A vowel in a word that

never exhibits vowel length would not be stored as long, since the learner would have no basis for storing it as such. The final two vowels in each of the words in (37) would be stored as long, since both appear at some point in the language as long vowels. I propose the underlying forms of words with long vowels in (38).

(39)	surface	underlying	gloss
	mem'de:wi	/memde:wi/	deer
	'me:hu	/me:hu:/	to be a boy
	'wa:šiw	/wa:ši:w/	Washo

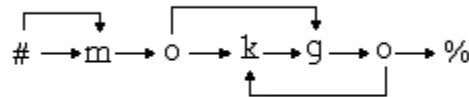
Since there is no evidence that the first vowel in *deer* is ever long, it is stored as short. The apparent vowel length shift is now simply analyzed as a WEAK PARADIGMATIC HEAD-DEPENDENT ASSYMMETRY (Dresher and van der Hulst, 1998) in which the head of a foot must be at least as complex as a dependent. The stressed syllable, the head can therefore be either long or short, but unstressed dependents may only be short.

Yu argues that the change in vowel length is a repair strategy for the fact that the stressed syllable is lighter than the unstressed syllable. His claim that syllable weight is the trigger for this repair seems unlikely, as there is little evidence that syllable weight is used in the stress calculation in Washo. His insight is, however, consistent with the head-dependent asymmetry approach. Although syllable weight does not affect stress calculation, a head may tolerate more complexity than a dependent, making the stressed syllable surface as heavier than the dependent when both are marked as long. Yu's claim that the stressed syllable is more complex than the unstressed is therefore correct; he errs in claiming that head calculation is a result of surface complexity instead of the converse.

#### 4.5 Linearization

As demonstrated above, using a directed graph as a representation of a word predicts the correct plural forms for almost all Washo examples. It is necessary, however, to explain how a speaker’s brain converts the data represented in a directed graph into a word, since a directed graph is an abstract data type, not a data structure with a predefined notion of direction and computation. Without some sort of clear process for converting directed graphs into words, a directed graph would predict any number of ungrammatical forms. If we define an acceptable word as any path through a directed graph that begins at # and ends at %, any of the following would be predicted to be acceptable by the graph in (40): mokgo (the singular form), \*mogo, mogokgo (the plural form), \*mokokogko, \*mogokgokgokgo.

(40)



It would, in fact, be possible to predict infinitely many words from this directed graph by repeatedly adding “kgo” to the end of the word by following the arcs between k and g, g and o, and o and k. The fact that this model would predict two acceptable forms (mokgo and mogokgo, the singular and plural of *shoe*, along with infinitely many unacceptable forms is clearly far from ideal. To prevent such rampant overprediction, it would be useful to have some limits on the tracing of arcs. I will call this process LINEARIZATION, as it is similar to linearization constraints in syntax (Koutsoudas, 1981).

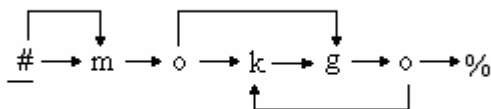
My study of directed graph representations of words is, for the most part, limited to this analysis of Washo plural reduplication, so I do not claim to propose a process of linearization that will produce the predicted output given any directed graph analysis of a phonological phenomenon. I propose instead some constraints of linearization that appear to be sufficient for the Washo example, and encourage further investigation of their applicability to other languages or processes.

I first propose that arcs bear some mark of their origin. That is, arcs that were part of the original stem and arcs added by morphological processes are respectively marked as such. I propose a linearization algorithm in (41). This algorithm would take as input a specific form with an attached directed graph structure. It is not equipped to assign a directed graph structure to a stem. I describe that morphological process involving the elsewhere condition in §4.1.3

- (41) Begin at #.  
 If there is a morphologically added arc, follow it and remove it from the arc set,  
       recording every sound that is traversed  
 If there is no morphologically added arc, follow a phonological arc, recording  
       every sound that is traversed  
 End at %.

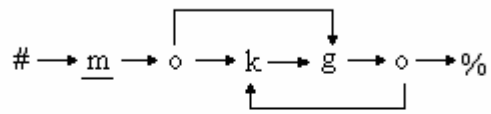
Consider an example of the execution of this algorithm in (42).

(42)

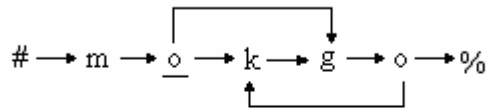


Start at #, take the available morphological arc, and delete it.

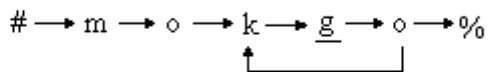




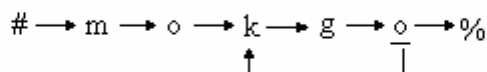
Record m, take the phonological arc, the only possibility  
word so far: m



Record o, take the available morphological arc and delete it  
word so far: mo



Record g, take phonological arc, the only one available  
word so far: mog



Record o, take the available morphological arc and delete it  
word so far: mogo



# → m → o → k → g → o → %

Record k, take the phonological arc, the only one available  
word so far: mogok



# → m → o → k → g → o → %

Record g, take the phonological arc, the only one available  
word so far: mogokg



# → m → o → k → g → o → %

Record o, take the phonological arc, the only one available  
word so far: mogokgo



# → m → o → k → g → o → %

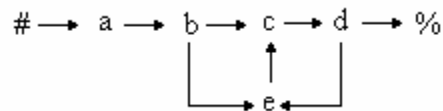
Algorithm terminates at %

At the termination of the algorithm, we are left with [mogokgo], the plural form of shoe.

There are a few essential features to this algorithm. The algorithm requires that there be a path to the end of the word that never gets deleted, since the algorithm cannot terminate until it reaches %. In the Washo reduplication case, that condition is always met, as the

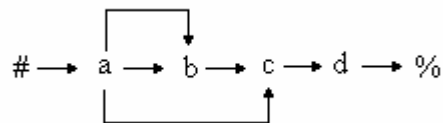
original phonological arcs must go to the end of the word and they cannot be deleted. Evidence against this algorithm would be a phonological process in some language in which a non-reduplicative affix were to be attached to a stem with multiple morphological arcs, such as the theoretical example in (43), where e is an infix, since the deletion of which could disrupt a path to %.

(43)



This sort of situation seems highly complicated and somewhat unlikely. The algorithm also requires DETERMINISM among the morphologically added arcs. That is, the algorithm has no provision for handling a case where two morphologically added arcs leave from the same segment, as there is no specified preference for taking one over the other. So this algorithm would not be able to run on a directed graph structure in which more than one possible path among the morphological arcs exists, such as the one in (44).

(44)



The demand for determinism is not a surprising requirement of an algorithm. In computing theory, an algorithm that can be computed deterministically is generally considered to be computable in a reasonable amount of time, while an algorithm that requires NONDETERMINISM, or the ability to make choices and run more than one path simultaneously, is generally computationally intractable. Examples of languages requiring nondeterminism among morphological arcs would not be particularly



surprising. A language with suffixing and reduplication occurring in the same word would be likely to exhibit this problem. To avoid the uncertainty introduced by nondeterminism, Raimy (2005) proposes a second type of marking, beyond phonological and morphological, in order to “break the tie” between two morphologically added arcs. While this idea is an interesting extension of the linearization algorithm proposed here, it is beyond the scope of this paper. The algorithm as it is presented is sufficient to handle Washo plural reduplication. I will leave possible extensions to those who are more versed in the technicalities of other examples.

The algorithm presented above is useful because it accurately accounts for all of the Washo data. It has, however, another advantage over some other systems of phonological computation. The directed graph analysis and the linearization algorithm are all computable in a reasonable amount of time. Computer scientists have a somewhat complicated definition of “reasonable” when it comes to the amount of time it takes to perform an algorithm. Algorithmically computational problems are divided into classes depending on how steps must be performed in order to solve them. The simplest classes are P and NP. While no one has managed to prove that P and NP are not equal, computer scientists would be highly surprised to find that they were. P stands for POLYNOMIAL TIME. If an algorithm is in the class P, the number of steps it takes to terminate in the worst case scenario is bounded by a polynomial (something like  $4n^2+3n$ , where  $n$  is the number of elements, in this case either sounds or arcs, in a specific instance of the problem). So the size of an instance of the problem, or the length a specific word that we would like to linearize, does have some effect on the amount of time the linearization algorithm will take to complete, but the number of steps does not grow unmanageably

quickly as the number of arcs increases. The number of steps is bounded by a polynomial. We can determine a polynomial bound by thinking about the worst case scenario: a template where a morphological arc leaves every segment. The number of required steps for the algorithm to terminate is therefore bounded by the square of the number of segments in the word, which is most definitely a polynomial.

Not all phonological computations can be performed in polynomial time. Some computations are in the class NP, or NONDETERMINISTIC POLYNOMIAL TIME. Algorithms in NP terminate in a polynomially bounded number of steps, but only when the machine is allowed to make choices. For an algorithm in P, the next step is determined by the algorithm. There is only one choice. But in NP, there could be many possible next moves at every decision point. If a computer (either an electronic one or a person who computes) attempts to follow the path of all possible decisions at the same time, it will obviously take longer than if only one path needs following. It is therefore impossible to simulate a nondeterministic process (assuming that P is unequal to NP) in deterministic polynomial time. The fastest known ways to simulate a nondeterministic polynomial time algorithm require exponential deterministic time - that is, the number of steps is an exponential expression like  $4^n$ . As is familiar to mathematicians, computer scientists, population biologists, or anyone else who has spent time studying exponential growth, this expression gets incredibly large extremely quickly. While  $2^4$  is only 16,  $2^{10}$  is 1024, and  $2^{15}$  is more than thirty thousand. A process that cannot be performed in deterministic polynomial time does not seem like a good cognitive model. The fact that the algorithm I present is in P is an important selling point, especially since Optimality Theory, the tool

used by Alan Yu in his analysis, has been proven to be NP-complete by Idsardi (forthcoming).

## **5. Conclusion**

There are many ways to measure the success of a linguistic analysis. The analysis of plural reduplication in Washo given in this paper accounts for nearly all of the available data. The few classes of words that are not fully explained by the analysis are likely lexical exceptions, as many appear to follow only slightly different rules. This paper also presents a consistent analysis, claiming that syllable weight is never a factor in prosodic calculations.

In this paper, I suggest that precedence variable calculation is the key to reduplication. While I make little reference to the calculation process, the variables I suggest are consistent with those used in many other prosodic calculations. I refer to onsets and vowels, as well as the first element, the last element, and an element following a vowel. All of these points are projected and calculated in attested SBG analyses of stress (Idsardi, 1992), tone (Prunell, 1998) (Kim, 1999), and other prosodic features. It therefore makes sense to refer to them in the calculation of reduplication.

Perhaps the most subtle success of this analysis is its failure on fundamentally unrelated forms. In most Washo plurals, an existing CV syllable is reduplicated. In words that lack such syllables, however, speakers must find another approach. A speaker who had heard only plurals of singular forms with CV syllables would be unable to guess with certainty the reduplication pattern of words without any onset. If an analysis is to model human language processing on some level, it ought to fail where a human being would be

unable to predict the reduplicated form. The analysis presented above for words with CV syllables refers to onsets. Clearly, a word without a CV syllable will not have an onset. We can therefore expect that our analysis will trigger an uncertainty, or another analysis. The attested forms demonstrate exactly that process. Plural forms of words that do not contain the necessary information to be explained by the first set of precedence variables can be analyzed by a similar and predictable, but fundamentally different directed graph structure.

In this paper, I also refine Rainy's directed graph method of phonological computation. I present a fuller analysis of the mathematics behind the model, as well as further discussion of its computational feasibility. Computation time is an interesting bridge between computer theory and linguistics, and deserves further exploration.

## **Bibliography**

- Broselow, E and J. McCarthy. (1983). A Theory of Internal Reduplication. *The Linguistic Review* 3:25-88.
- Dresher, Elan and Harry van der Hulst (1998). Head-dependent asymmetries in phonology: complexity and visibility. *Phonology* 15:317-352.
- Goodaire, Edgar and Michael Parmenter. (2002) *Discrete mathematics with graph theory*. Prentice-Hall, Inc. New Jersey.
- Harrison, David and Eric Raimy (2004). Harmony as Prosody. MOT Phonology Workshop. University of Ottawa.
- Idsardi, William J. (1992) The computation of prosody. Ph.D. dissertation. MIT.
- Idsardi, William J. (to appear) Calculating metrical structure. In Raimy, Eric and Charles Cairns (eds) *Contemporary views on Architecture and Representations in Phonological Theory*. MIT Press.
- Jacobsen, William H. (1964) A grammar of the Washo language. Ph.D. dissertation. University of California, Berkeley.
- Kim, Sun-Hoi. The Metrical Computation in Tone Assignment. Ph.D. dissertation, University of Delaware.
- Kiparsky, Paul (1973). Elsewhere in phonology. In Anderson, Stephen and Paul Kiparsky (eds.) *A Festschrift for Morris Halle*. New York: Holt, Rinehart and Winston.
- Koutsoudas, Andreas (1981). Constraints on Linearization Rules. *Language* 47:818-840.
- Marantz, Alec. (1984). Re Reduplication. *Linguistic Inquiry* 13:435-482.
- Purnell, Thomas (1998). Principles and parameters of phonological rules: Evidence from tone languages. Ph.D. dissertation, University of Delaware.
- Raimy, Eric (2000). *The phonology and morphology of reduplication*. Mouton de Gruyter.
- Raimy, Eric (2005). Personal Correspondence.
- Urbanczyk, Suzanne (1993). Infixation and moraic circumscription. UMOP 16.
- Winter, Werner. (1970). Reduplication in Washo: A Restatement. *International Journal of American Linguistics* 36:190-198.
- Yu, Alan (2003). The Morphology and Phonology of Infixation. Ph.D. dissertation, University of California, Berkeley.