Constraint-Ranked Derivation
A Serial Approach to Optimization

H. Andrew Black

SIL International

SIL e-Books

1

Library of Congress Catalog Number: 2006936601

ISBN-13: 978-1-55671-175-6

ISBN-10: 1-55671-175-3

ISSN: 1934-2470

Fair Use Policy

Series Editor
Mary Ruth Wise

Volume Editor
Mae Zook

Compositor
H. Andrew Black

Contents

ACKNOWLEDGEMENTS

I am indebted to a number of people, especially to my dissertation committee. The chair of the committee, Armin Mester, provided many fruitful hours of discussion throughout the development of this study. His insights, proddings, suggestions, criticisms, and encouragement were all invaluable. Junko Itô also afforded me many hours of profitable discussion. Jorge Hankamer contributed as well with his typical well-seasoned and thoughtful suggestions.

Discussions with Brian O'Herin, David and Judy Payne, Tom and Elizabeth Willett, and especially my wife, Cheri, also contributed to the ideas presented here.

I wish to thank my colleagues in SIL International for their moral support. Special gratitude goes to David Weber, Steve Echerd and the leadership of the Peru Branch who encouraged me to pursue this course of study.

Thanks are also due to the faculty of the Board of Studies in Linguistics at the University of California at Santa Cruz. All of the following have had a significant impact in my linguistic training: Judith Aissen, Chris Barker, Pratima Christdas, Sandy Chung, Jorge Hankamer, Junko Itô, Bill Ladusaw, Peter Lasersohn, Phil LeSourd, Jim McCloskey, Armin Mester, Jaye Padgett, Geoff Pullum, and Bill Shipley. The overall quality of teaching, love for the discipline, and ability to motivate and to develop quality from students has been superb.

I wish to give special thanks to my family which has been so patient and encouraging. My wife, Cheri, has been a true source of support. Our children, Kara, Lisa, Jeremy, and Matthew, have also been supportive, especially over the last three years when both Cheri and I have been graduate students.

This study program would not have been possible without the financial support, encouragement, and prayers of our many friends and supporters. I am also grateful for logistical support from the Linguistic Research Center, UC Santa Cruz.

This work is dedicated to the Source of all Life and Truth.

Great are the works of the Lord;
They are studied by all who delight in them.
Splendid and majestic is His work;
And His righteousness endures forever.
He has made His wonders to be remembered;
The Lord is gracious and compassionate.

Psalm 111:2–4

Chapter 1

Introduction

1.1 Overview

A familiar observation is that work in Generative Phonological theory over the past two decades has included a shift from a focus on sets of ordered rewrite rules to sets of well-formedness constraints (see e.g. Anderson 1985, Goldsmith 1990, McCarthy & Prince 1993:4–5). For example, the view that syllabification is rule-based (Kahn 1976, Steriade 1982, Levin(Blevins) 1985) has given way to the idea that syllabification is based on templates and well-formedness conditions (McCarthy 1979a, Selkirk 1982, Itô 1986, 1989). Such constraints provide an explanation for the types of changes that occur between underlying and surface forms. Epenthesis, to take an example, is no longer conceived of as a rule, but rather a result of the requirements of prosody (Itô 1989). A single well-formedness constraint may capture generalizations that required several distinct and seemingly unrelated rewrite rules. As usually conceived, an output representation must meet every relevant constraint in order to be considered well-formed. The nature of the interaction of these constraints, however, was left imprecise.

The emerging Optimality Theory (Prince & Smolensky 1991, 1992, 1993a, 1993b, McCarthy 1993, McCarthy & Prince 1992, 1993) provides a more rigorous treatment of constraint interaction. The constraints are ranked with respect to each other to provide a domination hierarchy. Crucially, lower-ranked constraints may be violated in order to meet higher-ranked constraints. The central idea is that the output of the grammar is the representation that best satisfies the set of ranked and violable

constraints; the output is the optimal representation, the best approximation to a target.

In traditional Generative Phonology, the production of the output representation is viewed as a sequential process. The underlying representation is modified (either by changing existing structure or by filling in required structure) in a stepwise fashion until the surface representation is produced.[1] In a striking departure from traditional Generative Phonology, Optimality Theory pursues an approach that does not involve rules or repair strategies. A generation function produces a potentially infinite set of candidate representations which are evaluated against the set of constraints. Rather than employing sequential rules or repair strategies to optimize an underlying form, a parallel "Best Satisfaction" algorithm is employed to select the optimal candidate from the set.

In setting out to do the research documented in this study, I wished to employ the insights of Optimality Theory to provide analyses for the truncation patterns in nominal stems in Southeastern Tepehuan and for the stress patterns of Pichis Asheninca. In addition, I wanted to submit the analyses to the rigors of a computational implementation.

Such a computational implementation has several benefits. First, it forces the analysis to be rigorous and precise. The details of every aspect of the analysis must be made explicit in order for the computer to know what to do. No 'hand-waving' is possible since computers are utterly unable to make the required assumptions.

Second, it provides a means for thoroughly and exhaustively testing a larger set of data. Every analysis is performed on a fixed set of data. By implementing an analysis,

[1] Earlier work (such as the foundational work of Chomsky & Halle 1968) sees the steps as deterministic. At each step in the derivation, it is precisely determined what the next step will be. There are no false steps; the output of rule R_i is always the input to rule R_{i+1}. Later work (such as the notion of Structure Preservation in Kiparsky 1982, among others) views the steps as nondeterministic. The output of a given rule might be blocked by a well-formedness constraint. The application of rule R_i might prove to be a false step, with the result that the rule does not apply.

one can quickly check the accuracy of the analysis on hundreds or thousands of forms. If a problem is thereby discovered, the analysis can be appropriately refined and then re-tested against the larger data base.

Third, if the tool is programmed appropriately, it provides a means for evaluating and doing further research. It allows the analyst to do more thinking and less time-consuming data checking. It also aids the researcher in exhaustively and incrementally analyzing a particular language. It is potentially the case that a particular analysis will work well for a subset of the data of a given language, but not necessarily for the entire language as a whole. Several such "independent" analyses may be proposed and individually tested. By implementing each one, the ramifications each has for the other can be quickly and relatively exhaustively determined. The empirical coverage of the analysis for the language as a whole is thereby more easily tested for consistency.

Three components would need to be implemented to computationally model Optimality Theory: (i) ranked and violable constraints (which evaluate phonological representations), (ii) a candidate set generator (which assigns structure to an underlying string in potentially infinitely many ways), and (iii) a "Best Satisfaction" evaluation procedure (which determines which candidate best satisfies the constraints).

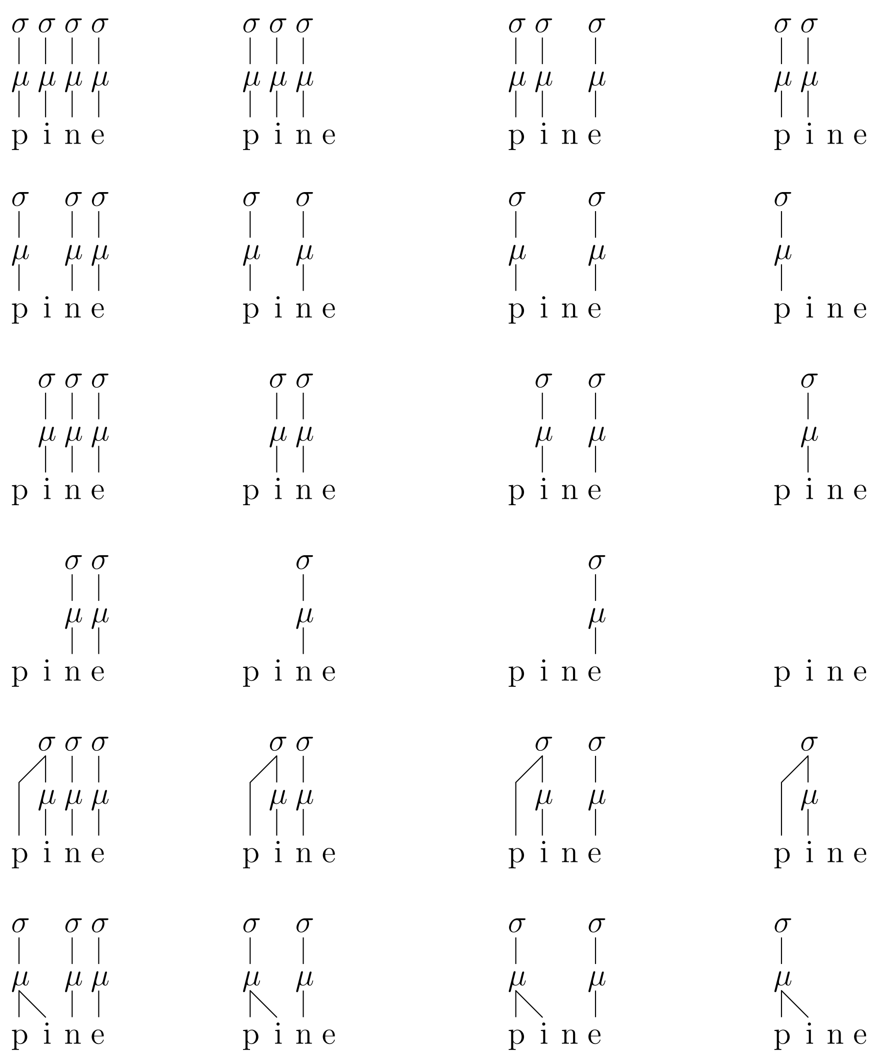
Phonological representations can clearly be implemented with appropriate data structures. Constraints can be viewed as functions which evaluate the representations and are, therefore, also implementable. The "Best Satisfaction" algorithm as formulated in either its perspicuous, but less formal manner in McCarthy & Prince (1992) or its more precise form in Prince & Smolensky (1993b:68-76) is also quite implementable. The generation function, however, presents some serious challenges to the implementor.

First, the infinite nature of the candidate sets presents potential processing difficulties. The generation of the set itself will never be completed. There will always be at least one more candidate to be added. Some kind of heuristic is required that

will limit a candidate set to a practical size without accidentally screening out the best (i.e. attested) candidate. If the only source of infinity within the candidate sets is due to epenthesis, then a simple heuristic such as disallowing more epenthesis sites than segments in the input string should limit the set appropriately.

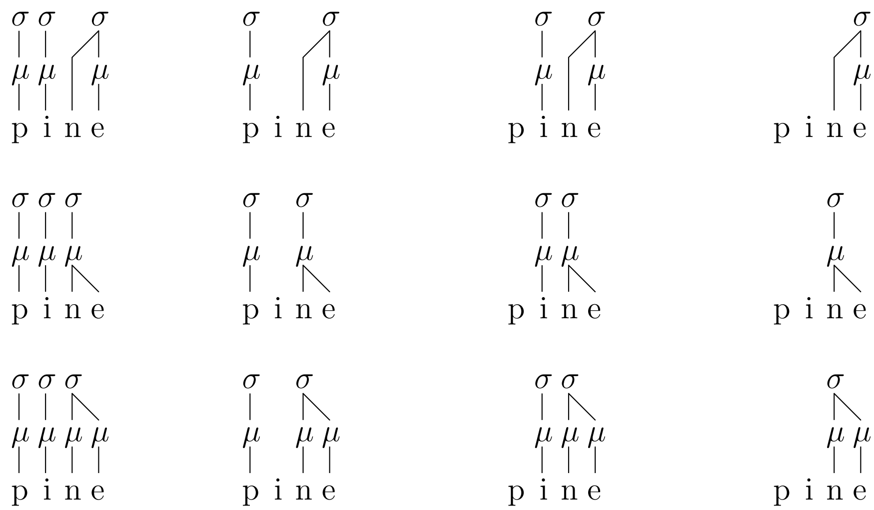
Infinity aside, even a short string such as pine will have a relatively large number of candidates as evidenced by the candidates listed in (1).[2]

(1) a. b.

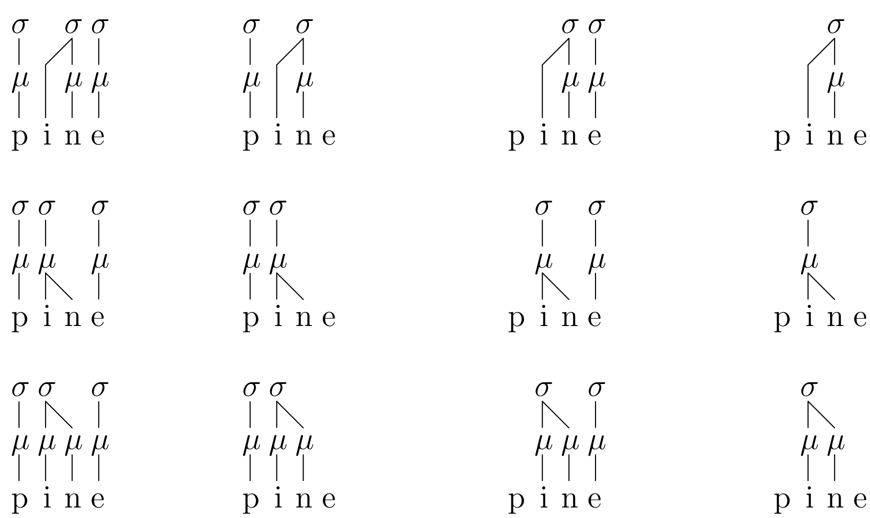


[2]Consonants must be able to bear moras in order to handle languages such as Berber, where any segment can be the nucleus of a syllable (Dell & Elmedlaoui 1985, 1988; see also Prince & Smolensky 1993b).

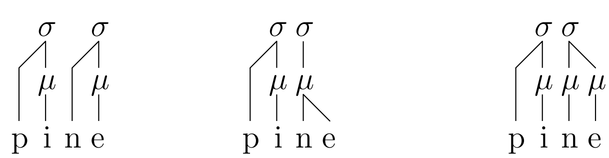
c.

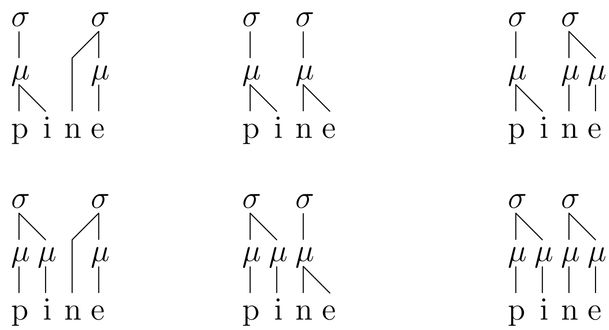


d.

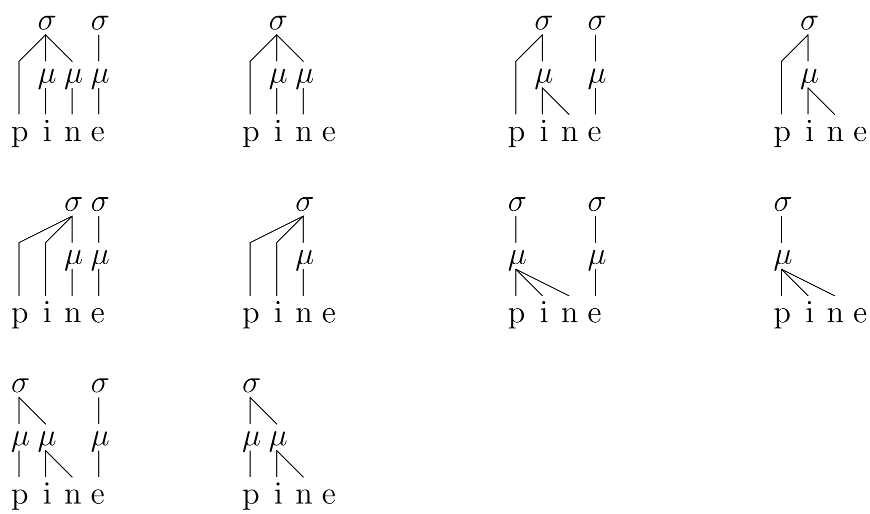


e.

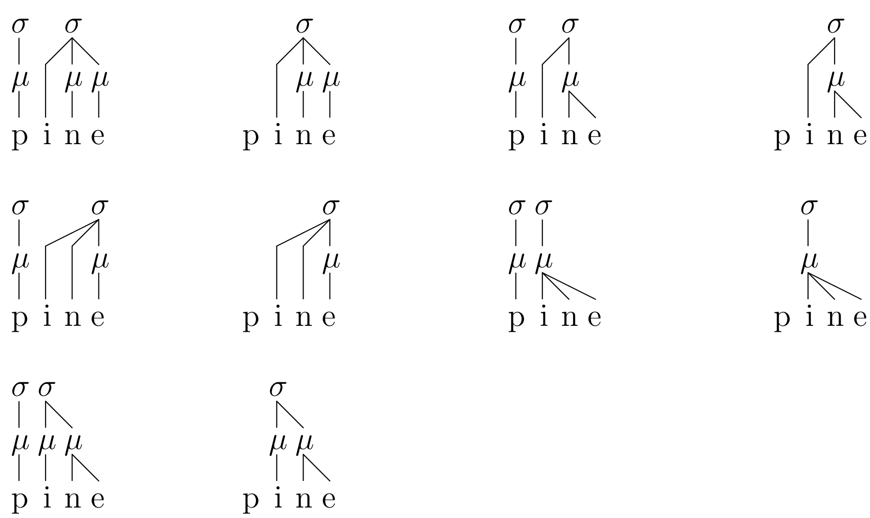


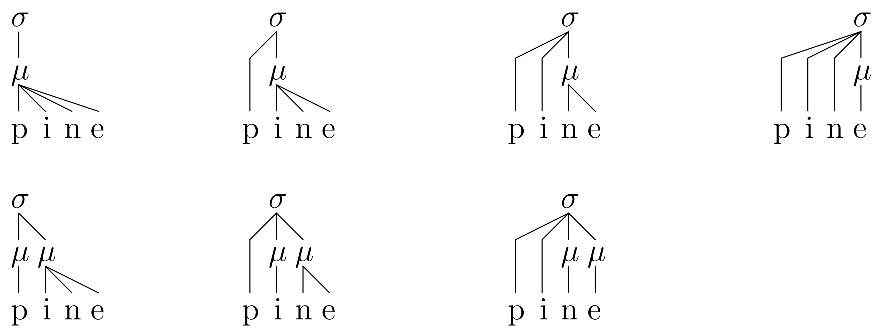


f.



g.



h. 

Each of the 88 candidates in (1) has an infinite number of other possibilities which include one, two, three, four, ... epenthetic segments. The delineation of the set of non-epenthetic candidates for the Turkish form in (2) is left as an exercise for the reader (the form is from Hankamer 1986:43).

(2) çöplüklerimizdekilerdenmiydi 'was it from those that were in our garbage cans?'

In addition, each such candidate would need to be footed in as many ways as theoretically possible. While implementing a generator for such sets may not be impossible, it is certainly not trivial.

Thirdly, the precise details of how the generator would create candidates with respect to segmental changes (such as spreading and assimilation) is not immediately clear (see Clements 1985, Sagey 1986, McCarthy 1988, Padgett 1991, among others for current accounts of such processes). How the generator would account for the interaction of underspecified underlying forms and their surface counterparts (Kiparsky 1982, Pulleyblank 1984, Archangeli 1984/88, Itô & Mester 1986, Steriade 1987, Clements 1988, Mester & Itô 1989, among others) is also not clear.

For these reasons, I have chosen to take another approach to optimization. McCarthy (1993) notes that the conception of optimization via ranked and violable constraints is logically independent of the notion of having the constraints select from a set of candidates. This study seeks to explore such an alternative to the parallel

candidate set approach which is eminently implementable. It follows a more conservative line which adopts the notion of optimization with respect to a constraint hierarchy while holding to the traditional conception of sequential application of processes. That is, it claims that phonology is modular: it requires optimization in terms of a hierarchy of ranked and violable constraints as well as rules and operations of the familiar kind. Furthermore, such familiar processes and optimization are not incompatible.

To blend optimization and processes, this approach provides an explicit algorithm to allow for ranked and violable constraints within the standard derivational model. It maintains the key notions relating to markedness of Optimality Theory while being demonstrably implementable.

The algorithm, called Constraint-Ranked Derivation, is introduced in section 1.3 of this chapter, along with a more thorough explication of Optimality Theory and the parallel candidate set approach. Chapter 2 applies Constraint-Ranked Derivation to the challenging truncation and reduplication processes among nominal stems in Southeastern Tepehuan. Besides demonstrating the viability of the Constraint-Ranked Derivation approach to optimization, it also includes an instantiation of the role of binarity in stem well-formedness. The Constraint-Ranked Derivation and parallel candidate set approaches are shown to fare equally well for this data. Chapter 3 examines the intricacies of stress in Pichis Asheninca within the Constraint-Ranked Derivation model. The analysis sheds light on the relation between footing and prominence. The chapter also discusses issues of optionality with respect to the Constraint-Ranked Derivation and the parallel candidate set approaches to optimization. Chapter 4 describes a computational implementation of Constraint-Ranked Derivation as applied to the Southeastern Tepehuan and Pichis Asheninca data.

Before proceeding to present Constraint-Ranked Derivation in section 1.3, I lay out some crucial theoretical assumptions in the next section.

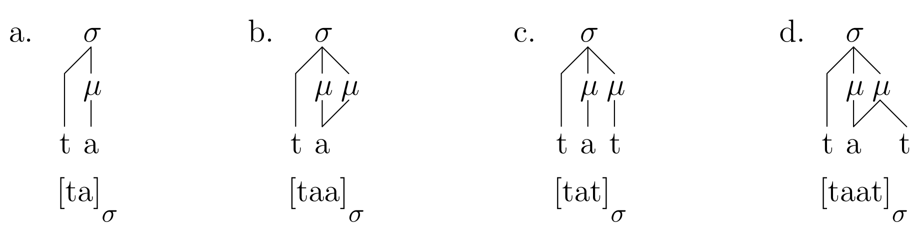
1.2 Theoretical Assumptions

This section makes explicit three areas of research on which this study depends. They will be adopted without argument. The first is a form of moraic theory which underlies the assumptions about syllable structure. The second relates to the prosodic hierarchy. The third is the use of repair strategies which is fundamental for the notion of Constraint-Ranked Derivation presented here.

1.2.1 Moraic Theory

The relationship between segmental material and prosodic structure has been argued to involve a prosodic tier consisting of moras (Hyman 1985, McCarthy & Prince 1986, Hayes 1989, Itô 1989). This accounts for syllable weight, issues of templatic morphology, compensatory lengthening and epenthesis in arguably better fashion than the ‘CV’ or ‘X’ tier view (McCarthy 1979b, Clements & Keyser 1983, Levin(Blevins) 1985).

Several versions of moraic theory have been proposed, especially in relation to whether or not onset segments are constituents of the mora (Hyman 1985, Itô 1989, Zec 1988, Hayes 1991, among others) or attach directly to the syllable node (Hayes 1989, McCarthy & Prince 1988, among others). While other versions are quite feasible, I will assume without argument that onset segments attach directly to the syllable node. Representative examples of the prosodic structure I will employ for light and various heavy syllables are given in (3).

(3)


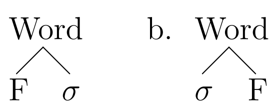
1.2.2 Weak Layering

Prososdic constituency consists of more than moras and syllables. Syllables are typically viewed as constituting feet and feet, words, etc. Selkirk (1984b:26) gives the "Strict Layer Hypothesis" which posits that "... a category of level *i* in the hierarchy immediately dominates a (sequence of) categories of level *i-1*." More recently, Itô & Mester (1992) have proposed Weak Layering. One of the key well-formedness conditions involved in Weak Layering is Proper Headedness, given in (4). This is coupled with a principle requiring maximal parsing.

(4) Proper Headedness (Itô & Mester 1992:12)
Every (nonterminal) prosodic category of level *i* must have a head,
that is, it must immediately dominate a category of level *i-1*.

Weak layering allows for structures such as those in (5), thereby avoiding positing degenerate feet (feet consisting solely of a light syllable $\breve{\sigma}$).

(5) a. b.



Such structures will play an important role in the analysis of the Southeastern Tepehuan data.

1.2.3 Repair Strategies

Constraint-Ranked Derivation employs the notion of a repair strategy (Paradis 1988, Yip 1988, McCarthy 1991, Mester 1994). According to Paradis (1988:71), a repair strategy is an operation applying to a representation that is triggered by the violation of some constraint. The result of such an operation 'repairs' the violation. No context is needed other than the constraint itself.

While phonological rules as conventionally conceived are arbitrary, repair strategies are related to well-formedness considerations. Repair strategies thus provide a

better explanation for why a particular modification to a representation exists. Further, by having the constraint determine the context of where the repair operation is to apply, repair strategies can be context-free. This avoids any problems of redundancy between the constraint and the context of the repair operation.[3]

As will become clear shortly, in Constraint-Ranked Derivation a constraint may have an associated repair strategy which is invoked whenever the constraint is violated. The repair strategy seeks to resolve the violation.

1.3 Introduction to Constraint-Ranked Derivation

As pointed out by Prince & Smolensky (1992), well-formedness constraints in phonological theory have often been used in a sub-formal manner. Optimality Theory (Prince & Smolensky 1991, 1992, 1993a, 1993b, McCarthy 1993, McCarthy & Prince 1992, 1993) provides a formal theory of constraint interactions, in which the constraints are violable and ranked in such a manner that, for a given input, they uniquely determine the optimal (i.e. the only well-formed) output of a grammar. The concepts of violability and ranking are both crucial. I will borrow these two notions without also employing Optimality Theory's non-derivational premise of candidate sets and how the constraint hierarchy is used to select the best candidate. Instead I will propose Constraint-Ranked Derivation which incorporates the optimization components of Optimality Theory while allowing for the sequential application of processes as in the traditional derivational model.

The two approaches will be illustrated by means of a simple example. Consider the verb base forms in (6) which are from Tiv (Arnott 1958, 1964, Pulleyblank 1984:217ff),

[3]Myers (1991) posits "persistent rules" to avoid such redundancies. As Armin Mester (p.c.) has pointed out, such rules seem to have a target, yet the rule is unable to express that target. The approach taken here using well-formedness constraints coupled with repair strategies is able to provide the target as well as the appropriate structural change.

an Atlantic-Congo language spoken in Nigeria (Grimes 1996:366). Certain verb tenses such as the Recent Past involve a process of ablaut as well as apocope. (Tone is ignored here.)

(6)

	Stem Form	Ablaut Form	Gloss
a.	pine	pin	'ask'
b.	gema	gem	'change'
c.	hura	hor	'weed'
d.	tsume	tsum	'make a mistake'
e.	yira	yer	'summon'
f.	hide	hidi	'return'
g.	venda	vende	'refuse'
h.	unde	undu	'mount'
i.	tɔngo	tenge	'blow (e.g. a flute)'
j.	dzɔhɔ	dzehe	'wrangle'

Pulleyblank (1984) notes that while the the Ablaut Form is predictable from the Stem Form, the converse is not. Ablaut will not be formulated here since we wish to focus on the vowel deletion phenomenon.

Note that the final vowel of the stem is truncated in the Ablaut Form for (6a–e) but is not for (6f–j). Apparently the final vowel of the Ablaut Form is deleted when it is identical with the preceding vowel and the intervening consonant is a sonorant. In Tiv only sonorants are allowed in coda position.[4]

A similar situation arises in Lardil (Hale 1973, Kenstowicz & Kisseberth 1979:112)

[4] This is a simplification. Voiced fricatives are also allowed in coda position. Arnott (1958:117) classifies these coda segments as voiced continuants.

Viewing the allowable codas from a sonority point of view, any segment can be a coda as long as its sonority is greater than or equal to that of a voiced fricative. Itô & Mester (1993) posit a well-formedness constraint dealing with the minimal sonority requirement for a mora as given in (i).

(i) Mora Sonority Threshold

$$\begin{array}{c} *\ \mu \\ | \\ [\textsc{sonority}< \alpha]_{\mathrm{Rt}} \end{array}$$

The value for α in Tiv would be '{voiced fricative}'. The constraint in (i) could potentially replace the CodaCond constraint in (7).

and also in Southeastern Tepehuan as chapter 2 will show. The underlying form for the ablauted (6a) is pini. Apocope successfully applies in a case like (6a), but is blocked by syllable well-formedness considerations in cases like (6f); the obstruent d cannot be licensed as a coda.

This section will motivate the relevant constraints to deal with the data in (6) in a step-wise fashion. The first sub-section illustrates how the data could be analyzed within Optimality Theory, including both the pertinent constraints and the parallel candidate set approach to optimization. The second sub-section then introduces the Constraint-Ranked Derivation approach to optimization.

1.3.1 Syllabification and Apocope in Optimality Theory

Since the success or failure of apocope in (6) is a function of syllable well-formedness, syllabification will be considered first.

1.3.1.1 Syllabification Constraints

The syllable well-formedness constraints are those typically associated with the process of syllabification (cf. McCarthy 1979a, Selkirk 1982, and Itô 1986, 1989 for example). Suppose the onset and coda constraints for Tiv are as in (7).[5]

(7)	Onset	A syllable must have one and only one onset consonant.
	CodaCond	Only a single sonorant consonant may occupy the coda position.

There are at least two other syllable well-formedness constraints.[6] The first, referred to as Parse, deals with prosodic licensing (Itô 1986, Goldsmith 1990, Itô & Mester 1993) and failure thereof ("stray erasure") (cf. McCarthy 1979b, Steriade 1982, Itô 1986, 1989, among others). Since Tiv consonants are apparently never deleted

[5]Onset here subsumes the complexity ban *Complex of Prince & Smolensky (1993b:87).

[6]In addition, there is -Cod: syllables do not have codas (Prince & Smolensky 1993b:34,85). For presentational perspicuity, I will leave out -Cod.

while vowels can be, Tiv requires two separate Parse constraints: one dealing with consonants and one with vowels.[7] These two constraints can be independently ordered within the hierarchy. The consonant-oriented constraint insists that all consonantal root nodes be syllabified. Such a consonantal melody preservation constraint is given in (8), where the vertical bar (|) indicates either direct or indirect domination.

(8) ParseC: $*\ \begin{array}{c}\sigma \\ \not| \\ \text{[+cons]}\end{array}$

That is, consonantal root nodes must be parsed into a syllable; they must be syllabified.

The vowel-oriented constraint is similar:

(9) ParseV: $*\ \begin{array}{c}\sigma \\ \not| \\ \text{[−cons]}\end{array}$

This insists that vocalic root nodes be parsed into a syllable. Recall that these constraints are violable; their relative ranking within the hierarchy determines how a particular form is optimized. This will become clearer below.

Optimality Theory has another syllable well-formedness constraint that addresses issues of epenthesis. As shown by Itô (1989), for those languages which display epenthesis, epenthetic segments are inserted in order to meet prosodic requirements. In the Optimality Theory model, the constraint Fill, informally given in (10), is ranked with respect to other constraints such as Onset, CodaCond, ParseC and ParseV to allow or disallow epenthesis.

(10) Fill: Unfilled syllable positions are prohibited.

[7] This separation of Parse into ParseC and ParseV is anticipated in Prince & Smolensky (1991) (see also Prince & Smolensky 1993b).

That is, syllables which have either an epenthetic onset or nucleus are to be avoided. Tiv does not seem to allow epenthesis of any kind.

Optimality Theory posits that these five constraints are ranked with respect to each other to account for the syllable patterns of a language. Following the notation of Optimality Theory, the ranking would need to be as in (11) for Tiv.

(11) $\textsc{Fill} \gg \left\{ \begin{array}{l} \textsc{Onset} \\ \textsc{CodaCond} \end{array} \right\} \gg \textsc{ParseC} \gg \textsc{ParseV}$

The $\gg$ symbol indicates a dominance relation; that is, FILL has a priority ranking over the other four constraints; both ONSET and CODACOND have a priority ranking over PARSEC and PARSEV; and, finally, PARSEC has a priority ranking over PARSEV.

FILL must be highest in order to account for the total lack of epenthesis: any epenthetic segment would immediately violate this highest constraint. The PARSEC constraint is ranked lower than either CODACOND or ONSET because it would negate the special limitations in these two constraints if it were ranked higher. For example, if PARSEC were higher than CODACOND, then it would insist that the d in the truncated string hid be syllabified even though this would violate CODACOND. Similarly, if PARSEC were ranked higher than ONSET, then the initial k in a string such as ksu would be forced to be syllabified as $[\text{ksu}]_\sigma$ in violation of the requirement that there be one and only one onset consonant. PARSEV is violated in apocope. Therefore, it must be ranked low.

Under the parallel candidate set approach to optimization, a given underlying form would have potentially infinitely many candidates which are evaluated against the constraint hierarchy to select the best one. The candidates are primarily variations on how the string of segments in the underlying form could be syllabified. Typically, all but one of these will be incorrect. The one candidate that best satisfies the ranking is selected as the output.

The algorithm employed to select the optimal candidate is given in (12).[8]

(12) Best Satisfaction (McCarthy & Prince 1992)

- Except for ties, the candidate that passes the highest ranked constraint is the output. (N.B., no trade-offs, negotiations, numerical optimizations.)
- A tie occurs when more than one candidate passes the highest ranked constraint; those that pass are tested recursively against the rest of the hierarchy.
- A tie also occurs when all candidates fail the highest ranked constraint; all candidates are then tested recursively against the rest of the hierarchy.

The parallel candidate set approach employs constraint tableaux to demonstrate the selection process. A schematic example is given in (14) (patterned after the one in McCarthy & Prince 1992). This example assumes that there are five constraints in the ranking given in (13).

(13) $C_1 \gg C_2 \gg C_3 \gg C_4 \gg C_5$

Only four of the potentially infinite candidates are shown and are labeled O_1 through O_4 (the 'O' is mnemonic for "option").

(14)

Candidates	C_1	C_2	C_3	C_4	C_5
O_1	*!				*
☞O_2		*		*	*
O_3		*	*!		
O_4	*!				

Whenever an asterisk (*) appears in row j, column k, it indicates that the candidate O_j violates constraint C_k. The asterisk with the additional exclamation mark indi-

[8]Prince & Smolensky (1993b:68–76) formalize this as Harmonic Ordering. The Best Satisfaction algorithm is used here for perspicuity.

cates a crucial failure: these candidates are definitely not the best and are eliminated from the set of potential candidates. Shading in a tableau element indicates that it is irrelevant whether the candidate passes or violates the constraint; the success or failure for the particular candidate has already been determined. The ☞ symbol indicates the candidate which best satisfies the ranked constraints; i.e. it is the output of the grammar.

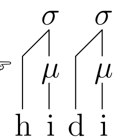
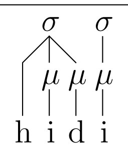
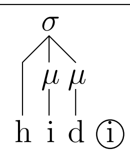
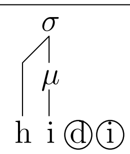
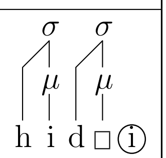
In (14), candidates O_1 and O_4 both violate constraint C_1. Because other candidates (here, O_2 and O_3) pass C_1, candidates O_1 and O_4 are eliminated from consideration. Since candidates O_2 and O_3 both pass constraint C_1, they tie and are recursively tested against the rest of the constraint hierarchy. They both violate constraint C_2, resulting in another tie. They are thus evaluated against constraint C_3. Only candidate O_2 passes constraint C_3. It is therefore chosen as the best candidate. Note that even though candidate O_2 violates constraints C_4 and C_5 while candidates O_1, O_3 and O_4 do not, it is still the best candidate given the hierarchy.

The tableau in (15) shows five of the potentially infinite syllabifications for the ablaut form of (6b).[9] The constraints are ranked as in (11) (with Onset arbitrarily ranked above CodaCond).[10]

[9]The potentially infinite number of candidates in the candidate set for syllabification has its source primarily in multiple instances of epenthetic segments.

[10]For formatting purposes, CodaCond is abbreviated as Coda in this and following tableaux.

(15)

Candidates	Fill	Onset	Coda	ParseC	ParseV
☞ 					
		*!	*		
			*!		*
				*!	*
	*!				*

An encircled letter indicates a deleted or unparsed segment. The box symbol (□) indicates an epenthetic segment. The first candidate passes every constraint. The second fails Onset because its second syllable lacks an onset. The third violates CodaCond since the coda consonant is non-sonorant. The fourth violates the parse constraints and the last one violates Fill.

1.3.1.2 Apocope

The tableau in (15), of course, does not consider the apocope process. For now, I posit that apocope is a constraint that disallows stem final light syllables (in the ablaut tenses).[11]

[11] That such apocope of the stem occurs in these particular tenses appears to be an idiosyncrasy of Tiv. In addition, the formulation of the constraint assumes that coda consonants resulting from

(16) APOCOPE: $*\breve{\sigma}]_{\text{Stem}}$

APOCOPE is violable: a surface form may violate APOCOPE if meeting APOCOPE would in turn violate one of the other higher-ranked constraints.[12] Consider the forms in (17), repeated from earlier.

(17)	a.	/pini/	⟶	pin	'ask'	(Ablaut form)	(=6a)
	b.	/hidi/	⟶	hi.di	'return'	(Ablaut form)	(=6f)

Both input forms contain a final light syllable. For (17b), however, if the final i̠ nucleus were to delete, then either the d̠ would become the coda of the preceding syllable or it would become unsyllabified and eventually stray erase. The first possibility would be a clear violation of CODACOND: non-sonorant d̠ is disallowed in coda position. The second would be a violation of PARSEC. This implies that APOCOPE must be ranked lower than CODACOND and PARSEC.[13]

What is the ranking between APOCOPE and PARSEV? APOCOPE in essence declares that a vowel in a final light syllable should not be parsed into a syllable, while PARSEV insists that every vowel be syllabically parsed. The more specific of the two must be ranked higher in order for it to be effective.[14] The ranking then becomes that of (18).

(18) $\text{FILL} \gg \left\{ \begin{array}{l} \text{ONSET} \\ \text{CODACOND} \end{array} \right\} \gg \text{PARSEC} \gg \text{APOCOPE} \gg \text{PARSEV}$

apocope will be moraic.

[12]It is logically possible for a representation that meets APOCOPE to also violate ONSET. The very nature of APOCOPE and ONSET, however, precludes this possibility: i.e. it is impossible for APOCOPE to produce a structure that would violate ONSET since APOCOPE only applies stem finally. By its very nature, a representation meeting APOCOPE may also violate PARSEV.

[13]For a language like Lardil, APOCOPE would be ranked above PARSEC (Prince & Smolensky 1993b:111).

[14]This is handled by Pāṇini's Theorem on Constraint-ranking (Prince & Smolensky 1993b:82) in Optimality Theory. See (190) in chapter 3 for further discussion.

The tableau in (19) shows how (17a) is selected using the more complete ranking of (18) (only five of the most relevant candidates are included).

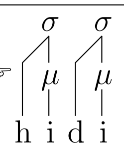
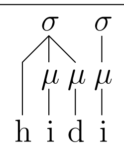
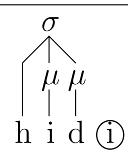
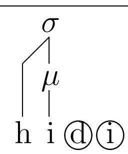
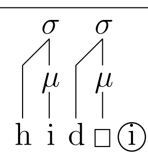
(19)

Candidates	FILL	ONSET	CODA	PARSEC	APOCOPE	PARSEV
σ σ μ μ p i n i					*!	
σ σ μ μ μ p i n i		*!			*	
☞ σ μ μ p i n ⓘ						*
σ μ p i ⓝ ⓘ				*!		*
σ σ μ μ p i n □ ⓘ	*!					*

The fifth candidate violates FILL and the second violates ONSET. These two candidates are thus eliminated from consideration. The n is a licit coda so the third candidate passes CODACOND. (The first and fourth candidates pass CODACOND vacuously.) PARSEC is violated by the fourth candidate, leaving only the first and third candidates. The first is eliminated by APOCOPE since it violates this constraint while the third candidate does not. Even though pin violates PARSEV, it is the optimal candidate.

The case of (17b) is shown in the tableau of (20).

(20)

Candidates	Fill	Onset	Coda	ParseC	Apocope	ParseV
☞ 					*	
		*!	*		*	
			*!			*
				*!		*
	*!					*

This tableau demonstrates that while hi.di violates Apocope (as indicated by the asterisk), the candidates hid and hi violate the higher-ranked constraints, CodaCond and ParseC, respectively. (They also violate ParseV). In addition, the second candidate crucially violates Onset and the final candidate violates Fill. The selected candidate is, of course, the attested form.

1.3.2 Syllabification and Apocope in Constraint-Ranked Derivation

As noted in section 1.1 above, it is not clear how to computationally implement the candidate set generation function. Is it possible, then, to maintain the precision

and insights of the ranking of violable constraints without employing parallel candidate sets? The answer to this question is in the affirmative. As also mentioned in section 1.1, McCarthy (1993) notes that the conception of ranked and violable constraints is logically independent of the notion of having the constraints select from a set of candidates. This has been illustrated by Mester (1994) for issues of Latin foot structure (although Mester 1994 includes discussion about the possibility of using a candidate set approach).

In this chapter, I propose an explicit algorithm which provides the details of how ranked and violable constraints can interact with representations within a derivational model. The algorithm has been successfully implemented (see chapter 4). The proposal makes use of many of the commonly assumed mechanisms. Crucially, it assumes that phonological representations are built by phonological processes. That is, a phonological process applies to a representation to produce a new representation. Constraints are well-formedness conditions that examine a representation; if the representation does not meet the well-formedness condition(s), then a repair strategy may be invoked (Paradis 1988, Yip 1988, McCarthy 1991). The expectation is that the repair strategy will modify the representation in such a way that it will then meet the well-formedness condition.

As mentioned earlier, the model I propose borrows several key concepts from Optimality Theory. In particular, the set of constraints are ranked with respect to each other and are violable. The representation that results from a process will be the best possible representation: that is, the representation will meet as many of the higher-ranked constraints as possible. Any constraint lower on the hierarchy must be met unless meeting it would lead to a violation of a constraint higher on the scale.

I will call this view Constraint-Ranked Derivation. It assumes the existence of a process P, an input representation R_{In}, and a constraint hierarchy H. A process applies to a representation to produce a representation which may be subject to a

particular set of well-formedness conditions. The set of well-formedness conditions may vary from process to process and a particular well-formedness condition may apply to more than one process.[15]

Constraint-Ranked Derivation can be formalized as in (21).[16]

(21)
$$< P : H > (R_{In}) \; = \; R_{Out}, \quad \text{where } H = C_1 \gg C_2 \gg \ldots \gg C_n$$

This notation indicates that the process P, subject to the constraint hierarchy H, applies to the input representation R_{In} to produce the output representation R_{Out}. The constraint hierarchy consists of n constraints ranked as in (21). In the limiting case, the hierarchy H is empty ($n = 0$). In this situation, the process is the same as a standard, exceptionless process or rule. The non-deterministic algorithm outlined in (22) explains how the process P is subject to the constraint hierarchy H.

---

[15] This is similar to how the set of constraints in Optimality Theory may vary from level to level and how a particular constraint may appear in more than one level (McCarthy & Prince 1993:24).

[16] Thanks are due to Armin Mester for suggesting this formalization which is patterned after the prosodic circumscription notation of McCarthy & Prince (1990).

(22) Constraint-Ranked Derivation Algorithm

1. Apply the process P to the input representation R_{In} to produce a tentative output representation R_{Out}.
2. Check the representation R_{Out} against each constraint in the hierarchy H, moving from the highest to lowest constraint (i.e. from C_1 to C_n).
3. As soon as a constraint (say, C_i) is violated:

3a. if C_i does not have an associated repair strategy, then

- 3ai. if no repairs have been attempted, the process blocks: use the original input representation R_{In} (i.e. $R_{Out} = R_{In}$).
- 3aii. if repairs have been attempted, constraint C_i is allowed to be violated; continue to check representation R_{Out} against the rest of the constraints in the hierarchy (i.e. C_{i+1} through C_n).

3b. if C_i has an associated repair strategy, apply that strategy to R_{Out} to produce a repaired representation R_R.

- 3bi. If R_R fails any constraints higher than or equal to C_i in the hierarchy (i.e. C_1 through C_i), the repair strategy blocks; continue to check representation R_{Out} against the rest of the constraints in the hierarchy (i.e. C_{i+1} through C_n). Constraint C_i is allowed to be violated.
- 3bii. If R_R passes all constraints higher than or equal to C_i in the hierarchy (i.e. C_1 through C_i), set R_{Out} to R_R and continue to check this repaired representation against the rest of the constraints in the hierarchy (i.e. C_{i+1} through C_n). R_{Out} becomes the optimal representation at this point in the hierarchy and is called the "representation of choice."

Step 3bi ensures that the repaired representation does not violate any higher-ranked constraint that was previously met. That is, it maintains the well-formedness of the representation with respect to the higher-ranked constraints; a lower-ranked constraint is not allowed to override a higher-ranked constraint. Step 3bii causes the representation to meet as many constraints as possible. A repaired representation is used whenever it improves well-formedness with respect to the hierarchy.

To illustrate how this works, consider the schematic derivation given in (24). The derivation delineates the steps for the formalization in (23).

(23) $<P : H> (R_0) \; = \; R_4,$
where $H = C_1 \gg C_2 \gg C_3 \gg C_4 \gg C_5 \gg C_6$

(24)

	Representation	Constraint	Repair	OK?
Input	R_0			
Process	R_1			
		C_1		
		$*C_2$	$\mapsto R_2$	$\surd$
	R_2	$*C_3$	$\mapsto R_3$	$* C_1$
		C_4		
		$*C_5$	$\mapsto R_4$	$\surd$
	R_4	$*C_6$	N/A	
Output	R_4			

We will assume that four of the six constraints, C_2–C_5, have associated repair strategies (this is not explicitly indicated in the derivation to avoid additional clutter). The other two, C_1 and C_6, do not have any associated repair strategies. The original input representation is R_0. The process applies to R_0 and produces the tentative representation R_1. R_1 is then checked against the ranked constraints. It passes C_1, but violates C_2 (as indicated by the asterisk next to C_2). Constraint C_2 has an associated repair strategy which is invoked to produce the repaired representation R_2 (as indicated by the 'maps-to' symbol $\mapsto$). R_2 is then checked against the higher or equal constraints C_1 and C_2. Since it passes both of these (as indicated by the $\surd$ in the "OK?" column), R_2 becomes the representation of choice. The next constraint, C_3, is then checked. R_2 violates C_3 so the repair strategy associated with C_3 is invoked and produces representation R_3. This must be checked against constraints C_1–C_3. For this illustration, assume that it violates constraint C_1 (as indicated by the asterisk followed by C_1 in the "OK?" column). As a result, R_3 is not used. R_2

is then checked against constraint C_4, which it passes. It violates C_5, though, and the repaired representation R_4 is produced. R_4 is then checked against the higher and equal constraints C_1–C_5. We will assume that it passes all of them. Finally, R_4 violates constraint C_6 which does not have any associated repair strategy. Since R_4 is now the representation of choice, it is still used.

As it stands, this scenario has a potential problem. There is no reason to assume that the repair to constraint C_5 will necessarily also repair the violation to constraint C_3. If it did not, then R_4 would violate C_3 and its repair would fail to take effect. This would make it impossible for constraint C_5 to have any effect at all, even if C_5 were totally independent of C_3.

There are at least two ways to solve this problem. One would be to mark all violated constraints as inactive. Repairs to lower-ranked constraints would then not be checked against such inactive higher constraints. The difficulty with this solution is that the repair for the lower-ranked constraint could introduce a violation of the higher-ranked constraint. If the higher-ranked constraint were marked as inactive, the violation would pass unnoticed.

Another way to solve this difficulty is to notice that constraints are typically concerned with the well-formedness of a particular domain. If a repair fails to improve the well-formedness of the domain, that particular domain can be marked as an exception to the constraint. Those domains which do meet the constraint (either initially or as a result of a repair) can then continue to be checked for well-formedness throughout the rest of the optimization. This is the approach taken here. See sections 2.3.4.1 and 4.2.3 for more discussion.

As will become clear in the following chapters, Constraint-Ranked Derivation maintains Optimality Theory's key insight of ranked and violable constraints while holding to the more traditional derivational model. It also maintains the "harmonic"

concept:[17] the new representation is submitted to the constraints in such a way that the "best" or most harmonic representation will result. The representation may be modified several times as it oscillates toward the optimal state.

One crucial difference between the two views is in how prosodic structure is assigned. Under the parallel candidate set approach, the mechanism that "generates" the candidate set assigns prosodic structure in every conceivable way to an underlying form (see (1) for an example). Syllabification oriented constraints such as ONSET, CODACOND, PARSEC and PARSEV are employed to select the most optimal candidate.

Contrastively, syllabification in the derivational approach assumed here is viewed as a persistent process (cf. McCarthy 1979a, Itô 1989:250). Syllabification involves the usual notions of building syllables maximally subject to the sonority hierarchy generalization and subject to language specific constraints such as ONSET and CODACOND.[18] Such process-internal constraints are not violable. They are not ranked and are not applied to the output of the process. Rather, they are an integral part of the process itself.

Under Constraint-Ranked Derivation, there would not be any need for FILL. In Optimality Theory, this constraint belongs to a class of constraints which serve to minimize some aspect of the ultimate representation. This class includes the constraint *STRUC: avoid structure (McCarthy & Prince 1993:22). *STRUC does not mean that structure is bad; to the contrary, it serves to help select the candidate with the least amount of required structure. Such structure-limiting constraints are

[17] Optimality Theory was originally called "Harmonic Phonology", then "Harmony-Theoretic Phonology" to reflect this notion, which stems from P. Smolensky's earlier work (Smolensky 1986). See Goldsmith (1990:323ff, 1991, 1993) for a different approach making use of the "harmony" concept in phonology.

[18] As noted by Inkelas (1989/90:271), maximal parsing essentially does the work of prosodic licensing. Prosodic licensing still exists and is applied to the output of processes in the form of ranked and violable constraints such as PARSEC and PARSEV. See (25) below for one such example.

crucial to any approach which selects among many candidates.[19]

Constraint-Ranked Derivation follows the traditional notions of structure being built by processes (such as syllabification and footing). Therefore, such constraints as FILL and *STRUC would not serve any useful purpose and are not included.[20]

Returning now to the Tiv example, under Constraint-Ranked Derivation, we can view the apocope as part of stem formation and formalize it as in (25).[21]

(25) $<\text{StemFormation} : H> (R_{in}) \;=\; R_{Out}$,
where $H = \text{PARSEC} \gg \text{APOCOPE} \gg \text{PARSEV}$

While PARSEC and PARSEV have no associated repair strategy, APOCOPE does. Mester (1994:14) posits that prosodic repair strategies fall into two classes as shown in (26).

---

[19] Itô (1989:243) notes that a principle that minimizes the number of syllables requires global computational power. Such a criticism could be leveled against the "Best Satisfaction" algorithm since it involves the computation of the minimal violation of structure-limiting constraints.

[20] This observation also serves to counter a potential objection to the Constraint-Ranked Derivation approach. (Thanks to Armin Mester for pointing out this potential problem to me.) Suppose a process is subject to a hierarchy consisting of two constraints, C_1 and C_2. In addition, suppose that C_1 is "Avoid X", where X is a basic structural element of a representation. Furthermore, suppose that C_2 has "Insert X" as its associated repair strategy. If C_1 is ranked above C_2, then the repair for C_2 would never be able to take effect: the repair would always constitute a violation of C_1.

For example, if the two constraints are FILLONS and ONSET, respectively, and ONSET has "Insert consonantal root node" as its repair, then the insertion of such a root node would always be a violation of FILLONS. (FILLONS minimizes onset epenthesis and is a necessary instantiation of FILL as is its counterpart FILLNUC; see Prince & Smolensky (1991) and Prince & Smolensky (1993b).) To say in such a case that the lower-ranked constraint C_2 has a repair strategy would be meaningless: it would never be able to repair anything.

This undesirable scenario is overcome by noting that no constraints of the type "Avoid X" with a corresponding "Insert X" repair strategy exist in the Constraint-Ranked Derivation approach. No such pairings as avoid/insert mora, avoid/insert syllable, avoid/insert foot, avoid/insert clash, avoid/insert root node, avoid/insert association line, or avoid/insert [nas] exist. This is because such "Avoid X" constraints are unnecessary.

[21] The apocope in Tiv must be viewed as relating to one or more constraints under the parallel candidate set approach because the only mechanisms available under this approach are constraints. Under the Constraint-Ranked Derivation approach, apocope in Tiv could be viewed as either due to a constraint or as a result of a process. The process could be $<P : H> (R_{In}) \;=\; R_{Out}$, where $P =$ "Delete a stem-final short vowel" and H consists solely of the constraint PARSEC. While the process approach might arguably be better, I have chosen to illustrate it here as a constraint in order to maintain a parallel between the two approaches.

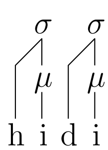
(26) Prosodic Repair Strategies (Mester 1994:14)

a. SHORTEN: REMOVE-μ
b. LENGTHEN: ADD-μ

These may be ranked with respect to each other on a language particular basis. The structural changes in (26) are invoked only when there is a violation of well-formedness. There is no redundancy between the statement of the well-formedness constraint and the environment of the rule that does the repair operation. For Tiv, I posit that the APOCOPE constraint pinpoints one place where well-formedness is violated in a representation. Tiv selects REMOVE-μ as its designated repair strategy.

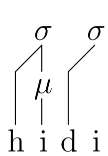
For example, consider the form hidi. Constraint-Ranked Derivation would proceed as follows. The syllabification process would create the representation for hidi as in (27).

(27)



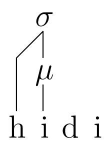
This representation is then submitted to the Stem Formation process which is subject to the constraints in (25). The representation satisfies PARSEC. APOCOPE, however, is violated, pinpointing the final mora as the locus for potential repair. REMOVE-μ is invoked and it delinks the final vowel from the prosodic structure to produce the tentative representation in (28).

(28)



Persistent syllabification then applies since prosodic structure has been modified. The final syllable no longer has a nucleus. Since Tiv has no epenthetic mechanism, the second syllable is removed to produce the tentative representation in (29).

(29)



Since the syllabification process is unable to place d as a coda (it is not a sonorant and thereby would violate CodaCond), the option with $[\text{hid}]_\sigma$ is never produced (it is rejected within the syllabification "box," so to speak).[22]

Each of the higher constraints is now applied in turn to the representation in (29). It violates ParseC, thus blocking the use of the final repaired representation in (29); that is, the original representation of (27) is employed. This means that Apocope is allowed to be violated. Since ParseV is lower than Apocope, it is then checked against the representation in (27). Since it meets ParseV, the representation in (27) is the final (attested) output.

This process is given schematically in (30).

(30)

	Representation	Constraint	Repair	OK?
Input	$[\text{hi}]_\sigma[\text{di}]_\sigma$	ParseC		
		∗Apocope	$\mapsto [\text{hi}]_\sigma\text{di}$	∗
		ParseV		
Output	$[\text{hi}]_\sigma[\text{di}]_\sigma$			

The derivation of the pin case of (6a) is given schematically in (31).

(31)

	Representation	Constraint	Repair	OK?
Input	$[\text{pi}]_\sigma[\text{ni}]_\sigma$	ParseC		
		∗Apocope	$\mapsto [\text{pin}]_\sigma\text{i}$	√
	$[\text{pin}]_\sigma\text{i}$	∗ParseV	N/A	
Output	$[\text{pin}]_\sigma$			

[22] Syllabification will not in turn moraify the final i (thus restoring the immediately preceding removal of that mora). As pointed out to me by Junko Itô, this can be viewed as a result of a slightly modified version of the Relinking Condition of Pulleyblank (1984:115).

The input consists of two light syllables as in the hidi case. ParseC is satisfied, but Apocope is violated and pinpoints the locus of repair. The designated repair for Apocope (remove-μ) removes the final mora. Syllabic structure has been modified, so persistent syllabification applies. Since the n is a sonorant, it can be syllabified as a coda. The representation resulting from this re-syllabification meets both the higher-ranked ParseC constraint and the Apocope constraint. This representation thus becomes the representation of choice. It is allowed to violate ParseV, yielding the attested output form, pin.

Notice that the set of representations produced during the derivation under the Constraint-Ranked Derivation model are precisely the most relevant candidates for the parallel candidate set approach.[23] In addition, the best representation in terms of the constraint hierarchy is output by both procedures. The results in this case are neutral between the two approaches.

Both the parallel candidate set approach and Constraint-Ranked Derivation involve procedures for determining the best or optimal representation given a hierarchy of constraints. Whereas the parallel approach keeps track of all possibilities at once, Constraint-Ranked Derivation pursues a sequential solution. Both involve non-determinism: more than one choice is often available. For the parallel candidate set approach, the "Best Satisfaction" algorithm whittles the candidates down from a potentially denumerably infinite set of choices to one candidate. Constraint-Ranked Derivation has at most two choices to select from at any one point in the derivation: it must decide between the current representation R_{Out} and the repaired representation R_R.

Constraint-Ranked Derivation, then, involves a minor amount of backtracking. This is a strategy for dealing with choices that pursues one possible choice while

[23] As Junko Itô (p.c.) has noted, this suggests that the Constraint-Ranked Derivation algorithm might be useful as a generation function for the parallel candidate set approach to optimization.

keeping track of the other possible choices so that if necessary it can go back and try the other possibilities later (see Winograd 1983:63 for a fuller explication). In the Constraint-Ranked Derivation case, at most one step will be backtracked. This is no more than has been posited for any kind of "rule blocking." For example, McCarthy (1986) argues that rules of syncope are blocked in Afar and Tonkawa precisely when syncope would produce a geminate and thereby violate the Obligatory Contour Principle (OCP). The Tonkawa data in (32) provide the key contrast.

(32)	a.	notoxo-	'to hoe'	notxoʔ	'he hoes it'
	b.	picena-	'to cut'	picnoʔ	'he cuts it'
	c.	hewawa-	'to die'	hewawoʔ	'he is dead'
	d.	ham'am'a-	'to be burning'	ham'am'oʔ	'he is burning'

The forms in (32a–b) undergo the pervasive syncope process in Tonkawa. Syncope fails to occur in cases such as (32c–d), where identical consonants would become contiguous if the intervening vowel were to be deleted. Since such contiguous identical consonants would constitute a violation of the OCP, the syncope process is said to be blocked by the OCP.

This means that the syncope rule would make one false step in the cases of (32c–d). In our terms, the original representation (R_{In}) would undergo the deletion, resulting in a modified representation (R_{Out}). R_{Out} would violate the OCP so the derivation would 'backtrack' and use the original representation (R_{In}). Notice that there is no globality involved here. The decision about which representation to use is always made locally.

The Constraint-Ranked Derivation model will be assumed in the discussion that follows. It will be further illustrated in later chapters containing case studies of nominal stem formation in Southeastern Tepehuan and stress assignment in Pichis Asheninca. Chapter 4 discusses the results of computationally implementing Constraint-Ranked Derivation and applying it to these case studies.

Chapter 2

Truncation and Stem Binarity in Southeastern Tepehuan

2.1 Introduction

The last chapter introduced Constraint-Ranked Derivation and its dependence on the notion of optimization from Optimality Theory. This chapter gives an analysis in terms of Constraint-Ranked Derivation for a fascinating truncation process among nominal forms in Southeastern Tepehuan, a Uto-Aztecan language spoken in the state of Durango, Mexico. Such truncation is often considered to be a result of syncope. Rules of syncope are not unusual among the world's languages. That unstressed vowels undergo syncope has been known for some time and is commonly assumed (Munro & Benson 1973, Kenstowicz & Kisseberth 1979:293,344, Bliese 1981:212–213, McCarthy 1986:220, Mester 1986:221, Halle & Vergnaud 1987b:28, Archangeli 1984/88:128ff among others). Piggott (1983:94) expresses this in terms of prosodic structure: only syllables in weak metrical positions undergo syncope. Both Halle & Vergnaud (1987b:29-30) and Hayes (1991:33–34) cite examples (including Tiberian Hebrew from Prince 1975) where stressed vowels do delete contrary to expectation. As they note, since the stress always shifts within the metrical foot, these cases still provide evidence for foot structure.

As will become clear below, while the usual syncope of vowels in weak position is an accurate description for many nominal stems in Southeastern Tepehuan, metrical position alone is not sufficient to explain all of the truncation patterns. The forms in

(33) illustrate how stem vowels in weak positions are deleted. The prosodic structure of the stem is given. Since stress assignment is iambic (see Section 2.2.3), iambic foot structure is built as indicated by the bracketing. Syllables with long vowels, diphthongs, and coda consonants all count as heavy. A period denotes a foot-internal syllable boundary and the stem is separated from any affixes by a dash. The right-hand column contains the surface form. Data are taken from E. Willett (1981b), (1982), (1985), and T. Willett (1991).[1]

(33)	a.	[taa]$_{\text{F}}$[ta.pɨɨ]$_{\text{F}}$si	$\longrightarrow$	[taat]$_{\text{F}}$[pɨš]$_{\text{F}}$	‘fleas’
	b.	[too]$_{\text{F}}$[si.ko]$_{\text{F}}$ri	$\longrightarrow$	[tooš]$_{\text{F}}$[koř]$_{\text{F}}$	‘pig’
	c.	[noo]$_{\text{F}}$[no.vi-d]$_{\text{F}}$	$\longrightarrow$	[noon]$_{\text{F}}$[vi’ñ]$_{\text{F}}$	‘his hands’
	d.	[tuu]$_{\text{F}}$[tu.vuu]$_{\text{F}}$[ri-d]$_{\text{F}}$	$\longrightarrow$	[tuut]$_{\text{F}}$[vuři’ñ]$_{\text{F}}$	‘his hips’

An apocope process deletes any word-final short vowel. The initial light syllable of a non-word-initial light-heavy ($\breve{\sigma}\bar{\sigma}$) foot undergoes syncope. The onset of the truncated syllable resyllabifies as the coda of the preceding syllable. Both the apocope and syncope processes arguably have a very reasonable phonetic explanation: word final vowels and vowels in the weak member of a foot are in phonetically weak positions.

Now consider the forms in (34).[2]

(34)	a.	[sui]$_{\text{F}}$[sui]$_{\text{F}}$[mari]$_{\text{F}}$	$\longrightarrow$	[suis]$_{\text{F}}$[mař]$_{\text{F}}$	‘deer (pl)’
	b.	hin-[nuu]$_{\text{F}}$[nuu]$_{\text{F}}$[tisV]$_{\text{F}}$	$\longrightarrow$	hiñ-[ñuun]$_{\text{F}}$[čiš]$_{\text{F}}$	‘my brothers-in-law’
	c.	[na.ka]$_{\text{F}}$[sɨ.rV]$_{\text{F}}$	$\longrightarrow$	[nak]$_{\text{F}}$[sɨr]$_{\text{F}}$	‘scorpion’

In (34a–b), the second syllable of the stem is heavy ($\bar{\sigma}$) and therefore would be the strong member of a foot. This is the syllable that is truncated, however. In addition, such a removal of a long vowel or diphthong is not expected as simply the result of

[1] Other work on this language includes that of E. Willett (1981a) and Kim (1987). The processes of palatalization and consonant mutation evident in this data are briefly discussed in section 2.2.1.

[2] As E. Willett (1982:169) notes, there are no underlyingly closed syllables in roots in Southeastern Tepehuan. Occasionally, then, it is necessary to posit an underlying vowel of unknown quality. Such a vowel is indicated by a small capital V in (34b–c) and other examples to follow.

syncope. Syncope in Tonkawa, for example, shortens long vowels; it does not remove them (Kenstowicz & Kisseberth 1979:431). In (34c), the metrically strong syllable of the initial foot is truncated. Clearly, more is involved than merely deletion of the nucleus of the weak syllable of a foot.

This chapter argues that the truncation patterns in Southeastern Tepehuan are a function of configurational rather than prominential considerations. The patterns can be explained by employing a set of well-formedness constraints, including a requirement that stems be binary at some level of prosodic structure (Itô & Mester 1992). Per the optimization notions of Optimality Theory, these constraints are ranked and violable. The analysis is presented primarily in terms of Constraint-Ranked Derivation, although an analysis in terms of the parallel candidate set approach to optimization is also discussed.

The chapter is organized as follows. Section 2.2 lays out the essential background information about Southeastern Tepehuan. Section 2.3 then provides the proposal for dealing with the data, including a ranking of the relevant constraints. Finally, section 2.4 addresses issues in reduplication in Southeastern Tepehuan. In particular, while previous treatments of reduplication in Southeastern Tepehuan lexically marked each form for whether it took a mono-moraic or bi-moraic template, this section demonstrates that the concept of binarity allows the template to be bi-moraic in the vast majority of forms.

2.2 Background Information

This section outlines the phonological inventory and syllable structure of Southeastern Tepehuan. The stress system of the language is also briefly outlined since this provides independent evidence of the foot structure operative in Southeastern Tepehuan.

2.2.1 Consonants and Vowels

The underlying consonant system is shown in (35).[3]

(35)

	Bilabial	Alveolar	(Alveo)palatal	Velar	Glottal
Nasal	m	n			
Stop	p, b	t, d		k, g	’
Fricative	v	s			
Liquid		r			
Approximant			y		h

Two pervasive allophonic processes are visible in the data. First, palatalization produces (alveo)palatals from the alveolars in the environment of the high front vowel i. These will be written as ñ, č, ǰ, š, and r̃.[4] Secondly, when voiced non-continuants are in coda position, they become preglottalized nasals at the same point of articulation. These will be written as ’m, ’n, ’ñ for b, d and ǰ, respectively. The velar g assimilates to the following consonant and becomes glottal stop word finally.

Southeastern Tepehuan has six vowels as shown in (36).[5]

(36)

	Front	Back Unrounded	Back Rounded
High	i	ɨ	u
Mid		ë	o
Low		a	

Vowels may be long or short. Diphthongs begin or end with a high vowel. The permissible diphthongs are ui, ɨi, oi, ai, io, ia, and ua.

[3]I have modified the system given in T. Willett (1991:13) slightly. I treat h as an approximant rather than a fricative since both h and y are the only illicit codas.

[4]The r̃ is phonetically a voiced palatal lateral fricative.

[5]This vowel system appears to be unbalanced with so many back vowels and only one front vowel. Perhaps the back unrounded vowels are actually central. I am presently not aware of any phonological reason for changing the system, however.

2.2.2 Syllable Structure

All syllables in Southeastern Tepehuan have one and only one onset consonant, including word initial syllables. There may be at most one coda consonant.[6] Syllables with a long vowel or a diphthong may also have a coda. All consonants except the approximants h and y are licensed in coda position. Using a template as a descriptive device, the syllable in Southeastern Tepehuan is CV(V)(C) with the second C and V optional.

2.2.3 Stress

The domain of stress is the stem in Southeastern Tepehuan. There are no secondary stresses. Iambic feet are built on the stem from left to right. As the forms in (37) show, the word stress is assigned via End Rule Left. (Stress is indicated by an acute accent over a vowel; non-stem material is enclosed in parentheses.)[7]

---

[6] A sequence of glottal-consonant is also allowed in coda position. There is always concomitant vowel rearticulation in such cases which I assume is due to a low level phonetic effect. No glottal-consonant sequences occur in any of the nominal stems I have found and, therefore, such sequences do not matter for our purposes here.

[7] Forms with two light syllables ($\breve{\sigma}\breve{\sigma}$), such as the adverbials mɨhɨ 'away (restricted opening)' and bɨhɨ 'toward (restricted opening)', do not bear stress (T. Willett, p.c. 1992). I know of no nominal or verbal stem which surfaces as $\breve{\sigma}\breve{\sigma}$; all underlying light-light ($\breve{\sigma}\breve{\sigma}$) stems surface as a closed heavy ($\bar{\sigma}$) due to truncation.

(37)
- a. tapɨ́ɨš ‘flea’
- b. takáarui’ ‘chicken’
- c. (hiš-)’ikóora’ ‘it’s dirty’
- d. mavíi’ñ ‘lion’
- e. i. (hiñ-)kóm ‘my back’
  - ii. komí(-’ñ) ‘his back’
- f. i. (hiñ-)húr ‘my heart’
  - ii. hurá(-’n) ‘his heart’
- g. yáatui ‘potato’
- h. vɨ́ɨpɨ’ ‘before’
- i. (hiñ-)ñúučiš ‘my brother-in-law’
- j. gáa’nga ‘looking for’

The forms in (38) demonstrate that diphthongs are bi-moraic since stem-initial syllables with diphthongs bear stress.[8]

(38)
- a. (hiš-)čáima’ ‘it’s fire-colored’
- b. nɨ́i’kar(-tam) ‘dancing place’
- c. gíotɨr ‘plain’
- d. vía’ñkai’ ‘lizard’
- e. ’óip(-imɨk) ‘they were walking around’
- f. báiǰa’ ‘fruit’

Only the stem bears stress; prefixes do not carry stress even when they form one or more feet as evidenced by forms like those in (39).

(39)
- a. (hiš-)čái(-ma’) ‘it is fire-colored’ (=38a)
- b. (vatu-)hugí(-a’) ‘he will now eat’
- c. (vahičuñ-)’áagi(-’ñ) ‘he then began speaking to me’

[8]Example (38b) is glossed as ‘dancing’ in E. Willett (1982:176). I assume this is a typographical error since it is listed as ‘dancing place’ in E. Willett (1981b:51). The -tam suffix is a postposition meaning ‘place (of)’ (E. Willett 1981b:51, T. Willett 1991:85).

2.3 Truncation and Constraint Interaction

Having laid out the essential background material, this section now turns to the analysis for the stem truncation patterns introduced in Section 2.1. The relevant constraints will be motivated in a step-wise fashion. The first sub-section addresses some issues relating to syllabification and apocope. Section 2.3.2 discusses some problems with a syncope-oriented analysis. The next section demonstrates how the notion of closed foot heads captures the patterns in the data, including the apocope process. Section 2.3.4 outlines the need for stem binarity and discusses the interaction it must have with the other constraints.

2.3.1 Syllabification and Apocope

The syllabification and apocope processes in Southeastern Tepehuan are quite similar to those for Tiv introduced in section 1.3.1.

2.3.1.1 Syllabification

For Southeastern Tepehuan syllabification, the onset and coda constraints as informally given in (40) are necessary.

(40)	ONSET	A syllable must have one and only one onset consonant.
	CODACOND	Any consonant except the approximants h̲ and y̲ may be in the one and only coda position.

No surface form ever violates either of these constraints since they are integral to the syllabification process.

As will become clear in the following, Southeastern Tepehuan consonants are never stray erased while vowels can be. Like Tiv, Southeastern Tepehuan requires that prosodic licensing of segments be broken into two separate PARSE constraints: one dealing with consonants and one with vowels. These two constraints can be independently ordered within a hierarchy and can apply to the output of any process

which may affect syllabic structure. The consonant-oriented constraint insists that all consonantal root nodes be syllabified.[9] Such a consonantal melody preservation constraint is given in (41) (repeated from chapter 1):

(41) ParseC: $*\ \overset{\sigma}{\underset{[+\text{cons}]}{\nmid}}$ (=8)

That is, consonantal root nodes must be parsed into a syllable; they must be syllabified.

The vowel-oriented constraint is similar (and is also repeated from chapter 1):

(42) ParseV: $*\ \overset{\sigma}{\underset{[-\text{cons}]}{\nmid}}$ (=9)

This insists that vocalic root nodes be parsed into a syllable. Recall that these constraints are violable; their relative ranking within the hierarchy determines how a particular form is optimized.

In passing, we note that Southeastern Tepehuan does not allow epenthesis of any kind.

2.3.1.2 Apocope

Southeastern Tepehuan displays what would traditionally be called an apocope rule. The forms in (43)–(44) show bi-syllabic and tri-syllabic apocope pairs, respectively. (The stem is separated from any prefixes or suffixes by a dash.)

[9] The processes of intervocalic h-drop and perfective formation do involve consonantal loss. From the derivational point of view, a distinction can be made between a (language specific) process and a constraint on a representation. Both intervocalic h-drop and perfective formation act on a representation to produce a new representation. Both the input and output representations must (and do) meet the requirements of the constraints.

(43) Bi-syllabic apocope alternations

a.	i.	/huutu-d/	⟶	huu.tu'n	'his fingernail'	
	ii.	/hin-huutu/	⟶	hiñ-huut	'my fingernail'	
b.	i.	/soiga-d/	⟶	soi.ga'n	'his pet'	
	ii.	/hin-soiga/	⟶	hiñ-šoi'	'my pet'	
c.	i.	/novi-d/	⟶	no.vi'ñ	'his hand'	
	ii.	/hin-novi/	⟶	hiñ-ñov	'my hand'	
d.	i.	/hura-d/	⟶	hu.ra'n	'his heart'	(=37f.ii)
	ii.	/hin-hura/	⟶	hiñ-hur	'my heart'	(=37f.i)
e.	i.	/toona-d/	⟶	too.na'n	'his leg'	
	ii.	/hin-toona/	⟶	hiñ-čoon	'my leg'	
f.	i.	/hogi-d/	⟶	ho.gi'ñ	'its hide'	
	ii.	/hogi/	⟶	ho'	'skin, hide'	

(44) Tri-syllabic apocope alternations

a.	i.	/hin-kusupa/	⟶	hiñ-ku.sup	'the back of my neck'	
	ii.	/kusupa-d/	⟶	kus.pa'n	'the back of his neck'	
b.	i.	/hin-noonovi/	⟶	hiñ-ñoo.nov	'my hands'	
	ii.	/noonovi-d/	⟶	noon.vi'ñ	'his hands'	(=33c)
c.	i.	/hin-tuvuuri/	⟶	hiñ-ču.vuuř	'my hip'	
	ii.	/tuvuuri-d/	⟶	tu.vuu.ři'ñ	'his hip'	

For now, apocope will be posited as a constraint that disallows word final light syllables.[10]

(45) Apocope: $*\breve{\sigma}]_{\text{Wd}}$

Apocope can be violated in Southeastern Tepehuan. A surface form may violate Apocope if meeting Apocope would in turn violate one of the other higher-ranked constraints. Consider the form in (46).

(46) /voohi/ ⟶ voohi 'bear'

[10] Apocope will be subsumed by another constraint in section 2.3.3.

This rare form[11] contains a final light syllable. If the i̲ nucleus were to delete, then either the h̲ would become the coda of the preceding syllable or it would become unsyllabified and eventually stray erase. The first possibility would be a clear violation of CodaCond; h̲ is disallowed in coda position. Persistent syllabification will not place h̲ as a coda. The second would be a violation of ParseC.

This implies that Apocope must be ranked lower than ParseC. The formalization then is as in (47).

(47) $<\text{StemFormation} : H> (R_{In}) \ = \ R_{Out}$,
where H = ParseC ≫ Apocope ≫ ParseV

While ParseC and ParseV have no associated repair strategy, Apocope does. For Southeastern Tepehuan, I posit that the Apocope constraint pinpoints one place where well-formedness is violated in a representation. Southeastern Tepehuan selects remove-μ as its designated repair strategy.

For example, consider the Southeastern Tepehuan forms in (48).

(48) a. /muuri/ ⟶ muuř̃ 'turtle'
b. /voohi/ ⟶ voo.hi 'bear' (=46)

For the normal case of (48a), Constraint-Ranked Derivation would proceed as in the derivation of (49). (These cases are quite similar to the Tiv cases of chapter 1.)

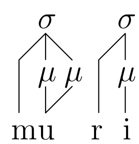
(49)

	Representation	Constraint	Repair	OK?
Input	$[\text{muu}]_\sigma[\text{ri}]_\sigma$	ParseC		
		*Apocope	↦ $[\text{muur}]_\sigma$ i	√
	$[\text{muur}]_\sigma$ i	*ParseV	N/A	
Output	$[\text{muur}]_\sigma$ i			

[11] This is the only instance of an Apocope violation I have found out of some 250 (non-compound) nominal forms other than its plural form, vapoohi. I have found two compound forms which violate Apocope: čiñvo 'mouth hair (beard or moustache)' from E. Willett (1981b:33) and kařĩ soi 'cricket' from Cervantes Solís, Ramírez Solís, Willett, & Willett (1991). Future research will hopefully shed light on how compounds are formed and why the Apocope violation should be allowed here (given that all underlying forms contain open syllables).

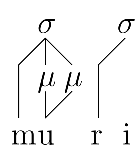
Initial syllabification would produce a representation for muuri as in (50).

(50)



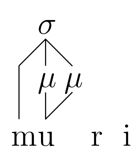
This representation is submitted to each of the ranked constraints of (47) in turn. The input passes the ParseC constraint, but violates Apocope. The latter constraint labels the final mora as the site of needed repair. The designated repair strategy (remove-μ) applies, removing the final mora to produce the tentative representation shown in (51).

(51)



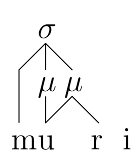
Since syllabic structure has been modified, persistent syllabification applies. The final syllable lacks a nucleus, so the offending syllable must be removed. The resulting tentative representation is given in (52).

(52)



Persistent syllabification then places the r as the coda of the remaining syllable producing the representation shown in (53). (The schematic version of this representation is displayed to the right of Apocope in (49).)

(53)



This representation satisfies both Apocope and the higher-ranked ParseC and thus becomes the representation of choice. ParseV is allowed to be violated since meeting

it would in turn violate the higher-ranked Apocope constraint. The result is the attested form, muur̃.

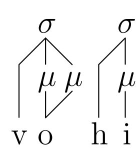
The process for the other case (48b) is given in (54).

(54)

	Representation	Constraint	Repair	OK?
Input	$[\text{voo}]_\sigma[\text{hi}]_\sigma$	ParseC		
		∗Apocope	$\mapsto [\text{voo}]_\sigma\text{hi}$	∗
		ParseV		
Output	$[\text{voo}]_\sigma[\text{hi}]_\sigma$			

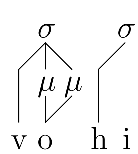
The syllabification process would create the representation for voohi as in (55).

(55)



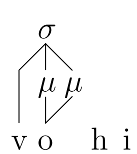
As for muuri, the representation satisfies ParseC, but violates Apocope. Remove-μ is invoked and it delinks the final vowel from the prosodic structure to produce the tentative representation in (56).

(56)



As before, persistent syllabification applies and produces the tentative representation in (57).

(57)



This time, the syllabification process cannot place h as a coda because it would be a violation of the CodaCond constraint. Each of the higher constraints is now applied

in turn to the representation in (57). It violates ParseC, thus blocking the use of the final repaired representation in (57). The original representation of (55) is employed instead with the result that Apocope is allowed to be violated. Because ParseV is lower than Apocope, it is then checked against the representation in (55). Since it meets ParseV, the representation in (55) is the final (attested) output.

2.3.2 Syncope Hypotheses

Ranking Apocope fairly high in the hierarchy will ensure that syllabification and foot building will produce structures without any final light syllables (with the exceptions mentioned above). How can we understand the other truncations that occur in Southeastern Tepehuan?

2.3.2.1 Weak Member of a Foot

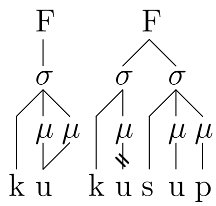
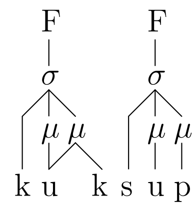
The syncope process could be hypothesized to be one which deletes the nucleus of the weak syllable of a foot. After all, such a position is phonetically weak; it lacks significant prominence. We could get this effect by making syncope be a constraint with the designated repair strategy remove-μ and ranking Apocope higher than Syncope.

The forms in (58) show how such a view of syncope would modify the results of apocope applying to the initial syllabification and foot structure.

(58)	a.	[noo]$_F$[no.vi'ñ]$_F$	⟶	[noon]$_F$[vi'ñ]$_F$	'his hands'	(=33c)
	b.	[too]$_F$[ši.kor̃]$_F$	⟶	[tooš]$_F$[kor̃]$_F$	'pig'	(=33b)
	c.	[goo]$_F$[go.goš]$_F$	⟶	[goo'ŋ]$_F$[goš]$_F$	'dogs'	
	d.	ha-[kuu]$_F$[ku.sup]$_F$	⟶	ha-[kuuk]$_F$[sup]$_F$	'the back of their necks'	

The hypothesized syncope of the stem in (58d) is illustrated in (59). Since the vowel u of the [ku]$_\sigma$ syllable is in the weak position within the foot, it would delete. Persistent

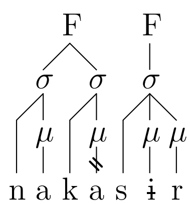
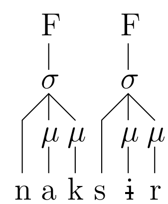
prosody (Hayes 1991:99) would then place the onset as the coda of the preceding syllable and would refoot as shown in (59b).

(59) a.  b. 

The difficulty with this syncope hypothesis is that it does not account for forms such as those in (60).

(60)	a.	[ku.su]$_F$[pa'n]$_F$	⟶	[kus]$_F$[pa'n]$_F$	'the back of his neck'	(=44a.ii)
	b.	[tɨ.ro]$_F$[vin]$_F$	⟶	[tɨr]$_F$[viñ]$_F$	'rope'	
	c.	[na.ka]$_F$[mɨr̃]$_F$	⟶	[nak]$_F$[mɨr̃]$_F$	'bat'	
	d.	[na.ka]$_F$[sɨr]$_F$	⟶	[nak]$_F$[sɨr]$_F$	'scorpion'	(=34c)

The truncation process is *not* merely deletion of the nucleus of the weak syllable of a foot: the nucleus of the strong syllable is deleted for the forms in (60). This is illustrated in (61) for (60d).

(61) a.  b. 

Notice how the nucleus of [ka]$_\sigma$ is deleted even though it is in the strong position of the foot. Southeastern Tepehuan, then, provides another case where a stressed vowel may delete.

Intuitively, it not hard to see why the weak member of the initial foot is not deleted in these cases. No onset clusters are allowed (*[ksu]$_\sigma$, *[tro]$_\sigma$, etc.) and ParseC is never violated (i.e. it is ranked high). Nonetheless, some explanation must be provided for why it is the second vowel that deletes in these cases.

2.3.2.2 Default Moraic Trochee

Can we save the hypothesis that the syncopated vowel is in a prosodically weak position by positing that in the case of two light syllables ($\breve{\sigma}\breve{\sigma}$), a default moraic trochee (McCarthy & Prince 1986:9) is built? Then for a case like (60a), the nucleus of the weak syllable of the foot [ku.su]$_{F_{mt}}$ would indeed delete to become the attested [kus]$_F$. The forms in (62), however, show that this cannot be correct.

(62) a. [naa]$_{F_i}$[na.ka]$_{F_{mt}}$[sɨr]$_{F_i}$ ⟶ [naan]$_F$[ka.sɨr]$_F$ 'scorpions'
b. [va.poo]$_{F_i}$[tV.po]$_{F_{mt}}$[da']$_{F_i}$ ⟶ [va.poot]$_F$[po.da']$_F$ 'worms'

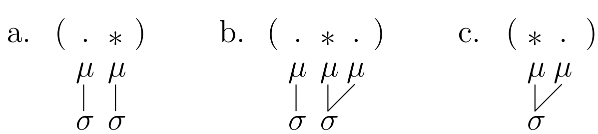
If moraic trochees were to be built here, then deleting the nucleus of the weak syllable of the foot would not produce the attested forms. For example, in (62a), the foot [na.ka]$_{F_{mt}}$ would become the unattested *[nak]$_F$.

Besides these empirical difficulties, this proposed solution would constitute a violation of the Uniformity Parameter (McCarthy & Prince 1986:10). This parameter requires languages to have the same foot type either for all words of the language or at least within a given word. A word is never allowed to mix foot types. The proposal would allow a single word to contain both iambic and trochaic feet as in (62b).

2.3.2.3 Syncope and Mora Prominence

If the truncation patterns cannot be explained by referring to the lack of prominence in the weak member of the foot, could the internal structure of the foot head be a factor? Prince (1983:57ff) noted a sonority decline in heavy syllables. Kager (1993:385) has updated this notion and assumes a moraic sonority decline as in (63) for the classic iambic foot types.

(63)



The most prominent mora in each foot is indicated by the asterisk (*). Mora prominence is foot final only in (63a), the light-light ($\breve{\sigma}\breve{\sigma}$) case. The other two cases, light-heavy ($\breve{\sigma}\bar{\sigma}$) and heavy ($\bar{\sigma}$), contain a final bi-moraic syllable. For these, mora prominence is not foot final.

An examination of the truncation data in Southeastern Tepehuan shows that almost no nominal forms end in a light syllable. In fact, there are no foot final light syllables anywhere within any stem. One might wonder, then, if this distribution is due to an avoidance of foot-final mora prominence. That is, if only (63b–c) were allowed in Southeastern Tepehuan, then not only would there be a prominence oriented explanation for truncation in Southeastern Tepehuan, but both apocope and syncope could be subsumed under mora prominence considerations.

While this might explain the apocope cases,[12] it fails to account for all of the syncope cases. Consider the forms in (64) which have the standard iambic foot parsing.

(64)	a.	$[\text{ku.su}]_F[\text{pa'n}]_F$	$\longrightarrow$	$[\text{kus}]_F[\text{pa'n}]_F$	'the back of his neck'	(=44a.ii)
	b.	$[\text{noo}]_F[\text{no.vi'ñ}]_F$	$\longrightarrow$	$[\text{noon}]_F[\text{vi'ñ}]_F$	'his hands'	(=33c)

Mora prominence would correctly predict the output in (64a): the foot $[\text{ku.su}]_F$ has final prominence which is to be avoided; by deleting the final vowel, the prominence shifts to the mora associated with the initial u̲. But for (64b), mora prominence predicts there should be no change at all. The second foot $[\text{no.vi'ñ}]_F$ meets the conditions of mora prominence. Since the initial vowel needs to delete, something other than mora prominence must be at play here. Unfortunately, then, mora prominence cannot be successfully invoked to subsume both apocope and syncope nor to explain the truncation patterns.

[12]It would also explain why all underlying light-light ($\breve{\sigma}\breve{\sigma}$) nominals surface as a single heavy ($\bar{\sigma}$).

2.3.3 Closed Foot Heads

Apparently, then, we must look beyond prosodic prominence for an explanation of the pattern in nominal stem truncations in Southeastern Tepehuan. An examination of the surface forms reveals that practically every surface nominal stem has two properties: (i) the head of every foot is closed whenever syllabically possible; and (ii) the stem is binary at some level of prosodic structure. That is, there are configurational well-formedness issues which drive the truncations.

In the sections below, the notion of a closed foot head will be explicated. Stem binarity will then be addressed in section 2.3.4.

2.3.3.1 Closed Foot Heads and Constraint-Ranked Derivation

As noted above in section 2.3.2.3, there are no cases of a surface form with a foot headed by a light syllable. The final syllable of every foot is always heavy and is often closed. This is not the first time such a configuration has been discussed in the literature. McCarthy & Prince (1990:240) note the existence of a lexical requirement in Arabic for all stems to end in a consonant. In addition, the dialect of English spoken in Boston and surrounding communities involves "r-intrusion" as illustrated in (65).[13]

(65) a. The spa<u>r</u> is broken.
b. He put the tuna<u>r</u> on the table.
c. The boat tends to yaw<u>r</u> a little.

The intrusive *r* is underlined. McCarthy (1993) argues that the presence of the intrusive *r* is the result of a ranked and violable constraint prohibiting a final (short) vowel in a prosodic word.

[13] The examples are from McCarthy (1993). See the references cited therein for other work on this dialect.

A similar constraint will capture the observations for Southeastern Tepehuan. The constraint in (66) is patterned after the one given by McCarthy (1993) for Boston English.

(66) ClosedFtHd: $*\ \text{V}]_{\text{Ft}}$

That is, prefer a closed foot head (one that ends in a consonant). (More precisely, if a constituent is a foot, then it should end in a consonant.) The associated prosodic repair strategy is the designated one for Southeastern Tepehuan: REMOVE-μ.

Before placing ClosedFtHd in the constraint hierarchy, we note that Apocope is subsumed by ClosedFtHd. To see this, consider the forms in (67), repeated from earlier.

(67)	a.	/tɨ.ro.vi.nV/	⟶	$[\text{tɨr}]_{\text{F}}[\text{viñ}]_{\text{F}}$	‘rope’	(=60b)
	b.	/na.ka.mɨ.ri/	⟶	$[\text{nak}]_{\text{F}}[\text{mɨr̃}]_{\text{F}}$	‘bat’	(=60c)
	c.	/na.ka.sɨ.rV/	⟶	$[\text{nak}]_{\text{F}}[\text{sɨr}]_{\text{F}}$	‘scorpion’	(=34c)

If Apocope is a distinct constraint, then there must be two steps: (i) the final light syllable violates Apocope with the result that the final mora is removed, and (ii) ClosedFtHd is violated by the initial foot, resulting in the loss of the final mora in that foot. The Apocope constraint, though, is redundant for these cases.

Apocope effectively causes the preceding syllable to become closed as illustrated schematically in (68) for the four possible cases where apocope could apply. (Recall that all underlying forms in Southeastern Tepehuan consist of sequences of open syllables.) The result is always the closing of a foot.

(68)	a.	[cv.cv] #	⟶	[cvc] #
	b.	[cv.cv]cv #	⟶	[cv.cvc] #
	c.	[cvv]cv #	⟶	[cvvc] #
	d.	[cv.cvv]cv #	⟶	[cv.cvvc] #

In (68a), the final light syllable is the head of the final foot. When the final syllable loses its mora, the onset is syllabified as the coda of the preceding syllable. The end result is that the foot head is closed. In (68b), the final light syllable is not footed. The deletion of its mora also results in the onset becoming the coda of the preceding syllable. The same result obtains for (68c–d).

Apocope, then, is not needed as a separate constraint. We replace Apocope with ClosedFtHd in the hierarchy yielding the ranking in (69).

(69) ParseC ≫ ClosedFtHd ≫ ParseV

The derivation in (70) illustrates how this works for (67b).[14]

(70)

	Representation	Constraint	Repair	OK?
Input	[na.ka]$_F$[mɨ.ri]$_F$	ParseC		
		∗ClosedFtHd	↦ [nak]$_F$a[mɨr]$_F$i	√
	[nak]$_F$a[mɨr]$_F$i	∗ParseV	N/A	
Output	[nak]$_F$a[mɨr]$_F$i			
Surface	[nak]$_F$[mɨr̃]$_F$			

The initial syllabification and footing is given in the input line. This satisfies ParseC because every consonant is syllabified. ClosedFtHd is violated by both feet. The designated repair remove-μ applies to close both feet as shown to the right of ClosedFtHd. While the resulting representation violates ParseV, ParseV is allowed to be violated because ClosedFtHd is ranked higher than ParseV. The

[14]A full derivation consists of several processes being applied to a representation according to the algorithm outlined in (22) of section 1.3.2. For the sake of exposition, I will assume that the allophonic modifications observed in surface forms (such as palatalization and pre-glottalization) occur sometime after the output of this evaluation. (In particular, the initial consonant of a stem may become palatalized due to the presence of a high-front vowel in a prefix; see (43b-ii,c-ii,e–i), among others.) The crucial point is that even an i̠ which has lost its mora will cause palatalization so this segment must be around before palatalization occurs. Segments which have lost their moras due to the remove-μ repair strategy are retained to allow for the segmental modifications which do occur in Southeastern Tepehuan. That is, stray erasure is assumed to occur after the palatalization operation.

surface form reflects the palatalization of the final consonant brought about by the final, unparsed i.

The cases in (71) will further illustrate the coverage of ClosedFtHd.

(71)	a.	/noo.no.vi-d/	⟶	[noon]$_F$[vi'ñ]$_F$	'his hands'	(=33c)
	b.	/too.na-d/	⟶	[too]$_F$[na'n]$_F$	'his leg'	(=43e.i)
	c.	/baa.banV/	⟶	[baa]$_F$[ban]$_F$	'coyotes'	

The derivation for (71a) illustrates a case where the final foot head is initially closed, but the initial foot head is not.

(72)

	Representation	Constraint	Repair	OK?
Input	[noo]$_F$[no.vid]$_F$	ParseC		
		*ClosedFtHd	↦ [noon]$_F$o[vid]$_F$	√
	[noon]$_F$o[vid]$_F$	*ParseV	N/A	
Output	[noon]$_F$o[vid]$_F$			
Surface	[noon]$_F$[vi'ñ]$_F$			

ClosedFtHd seeks to close the head of each existing foot. The only possible way to do this for the initial foot is to target the non-head of the following foot as the site for repair. remove-μ applies to the following light syllable (no) and persistent prosody is able to syllabify the n as a coda. The second foot also is able to remain even though it loses its weak syllable. In closing heads of feet, the site of repair will either be the nucleus of a following light syllable or the head of the foot.

This restructuring of prosodic material might appear to constitute a violation of the Free Element Condition of Prince (1985) given in (73) (see also Steriade 1988 and Halle & Kenstowicz 1991).

(73) Free Element Condition (FEC) (Prince 1985)

> Rules of primary metrical analysis apply only to Free Elements—those that do not stand in the metrical relationship being established; i.e. they are "feature-filling" only.

The FEC governs 'free' elements, those which have not been previously incorporated into prosodic structure. The restructuring that occurs in the cases studied here is always after primary foot construction. Initial footing does indeed meet the FEC: when feet are originally constructed (as depicted in the 'Input' line of the derivations), existing foot structure is respected. Under the analysis posited here, foot construction for Southeastern Tepehuan is quite straightforward. The perturbations in foot structure are a result of overriding configurational requirements that result in modifications of primary prosodic structure.[15]

Turning now to the example in (71b), note that the head of the initial foot is not closed. The derivation is given in (74).

(74)

	Representation	Constraint	Repair	OK?
Input	$[\text{too}]_F[\text{nad}]_F$	ParseC		
		*ClosedFtHd	↦ $[\text{toon}]_F\text{ad}$	*
		ParseV		
Output	$[\text{too}]_F[\text{nad}]_F$			
Surface	$[\text{too}]_F[\text{na'n}]_F$			

ClosedFtHd identifies the initial foot as lacking a closed head. The attempted repair, however, leaves the final consonant unparsed.[16] The higher-ranked ParseC constraint is thereby violated with the result that the repaired representation is rejected. ClosedFtHd is allowed to be violated in order to avoid violating a higher-ranked constraint.

The example in (71c) demonstrates the need for a directional-iterative application of repair. Both feet lack closed heads. Suppose the repairs are applied in a right-to-left direction. Such a derivation would be as in (75).

[15] Chapter 3 contains an account of stress in Pichis Asheninca where footing is also straightforward, but prominence is perturbed in accordance with prominential well-formedness requirements.

[16] The sequence ad would not be syllabified for two reasons: (i) the a is no longer moraic and (ii) even if a were moraic, $*[\text{ad}]_\sigma$ would violate the Onset constraint.

(75)

	Representation	Constraint	Repair	OK?
Input	[baa]$_F$[ba.nV]$_F$	ParseC		
		*ClosedFtHd	↦ [baa]$_F$[ban]$_F$V	√
	[baa]$_F$[ban]$_F$V		↦ [baab]$_F$anV	*
		*ParseV	N/A	
Output	[baa]$_F$[ban]$_F$V			
Surface	[baa]$_F$[ban]$_F$			

The ClosedFtHd constraint identifies both feet as lacking closed heads. If the final foot undergoes repair first, then the resulting representation is as shown to the right of ClosedFtHd in (75). Since the repaired foot now satisfies ClosedFtHd as well as all higher-ranked constraints, the representation becomes the representation of choice. The initial foot is then targeted for repair. The resulting representation, though, violates ParseC and is rejected. (The final vowel has lost its mora and therefore cannot constitute the nucleus of a syllable; the n cannot thus be parsed as an onset.)

If the repairs were to be applied in a left-to-right fashion, the results would be as in (76).

(76)

	Representation	Constraint	Repair	OK?
Input	[baa]$_F$[ba.nV]$_F$	ParseC		
		*ClosedFtHd	↦ [baab]$_F$a[nV]$_\sigma$	√
	[baab]$_F$a[nV]$_\sigma$	*ParseV	N/A	
Output	[baab]$_F$a[nV]$_\sigma$			
Surface	*[baab]$_F$[nV]$_\sigma$			

Again, both feet would be targeted for repair. The repair of the initial foot would cause the removal of the final foot as a side-effect. Therefore, no other repair would

be performed. The result, though, is incorrect.[17]

Why should the repair proceed from the right toward the left? One possibility is that it reflects the nature of the ClosedFtHd constraint. The constraint asserts that the right edge of a foot should be a consonant. If the focus is the right edge, then an iterative sweep of repair application from the right would be appropriate.[18]

2.3.3.2 Closed Foot Heads and Parallel Candidate Sets

Before turning to the role of Stem Binarity, we consider how the parallel candidate set approach to optimization would fare with the constraints in (69). The forms in (77) will show some crucial distinctions between the candidate set approach and the Constraint-Ranked Derivation approach.

(77)	a.	/hin-kusupa/	⟶	hiñ-$[\text{ku.sup}]_F$	'the back of my neck'	(=44a.i)
	b.	/baa.banV/	⟶	$[\text{baa}]_F[\text{ban}]_F$	'coyotes'	(=71c)

The candidate set tableau in (78) shows the results for the most relevant prosodic parsings of the stem in (77a).

[17]Pursuing the idea suggested by Junko Itô (p.c.) that the Constraint-Ranked Derivation algorithm might be useful as a generation function for the parallel candidate set approach (see footnote 23, page 31), we note that such a generator would need to apply repairs in both a right-to-left and a left-to-right direction.

[18]The requirement that feet be built from left-to-right coupled with the directionality of repair from right-to-left demonstrates the difficulty of attempting to build appropriately closed feet during initial footing. Such footing would need to have sufficient globality to be able to look ahead far enough to determine which vowel to skip. Compare the formation of the initial foot in the two forms in (i).

(i)	a.	/na.ka.mɨ.ri/	⟶	$[\text{nak}]_F[\text{mɨr̃}]_F$	'bat'	(=60c)
	b.	/ka.ka.rV.va.si/	⟶	$[\text{ka.kar}]_F[\text{vaš}]_F$	'goats'	

Whether the vowel of the second or third syllable of the underlying form would be skipped depends on the nature of the following syllables. The analysis proposed here has footing and syllabification respecting underlying segments without any recourse to globality. Thus, initial syllabification and footing is simple and intuitive.

(78)

Candidates	ParseC	ClosedFtHd	ParseV
[ku.su]$_F$[pa]$_\sigma$		*!	
☞[kus]$_F$ⓤ[pa]$_\sigma$			*
☞[ku.sup]$_F$ⓐ			*

The result is indeterminate between the last two. Both meet ClosedFtHd; both have a single violation of ParseV.

The form in (77b) is even more troublesome as shown in the tableau of (79).

(79)

Candidates	ParseC	ClosedFtHd	ParseV
[baa]$_F$[ba.nV]$_F$		**!	
[baa]$_F$[ban]$_F$Ⓥ		*!	*
☞[baab]$_F$ⓐ[nV]$_\sigma$			*

The optimal form with respect to this hierarchy is the unattested [baab]$_F$[nV]$_\sigma$. While the correct form has one violation of ClosedFtHd, the selected one does not have any (the one and only foot head is closed).

To resolve this problem, another constraint could be introduced such as Apocope. As shown in (80), this would select the correct [ku.sup]$_F$ over the unattested [kus]$_F$[pa]$_\sigma$.

(80)

Candidates	ParseC	Apocope	ClosedFtHd	ParseV
[ku.su]$_F$[pa]$_\sigma$		*!	*	
[kus]$_F$ⓤ[pa]$_\sigma$		*!		*
☞[ku.sup]$_F$ⓐ				*

Apocope is able to make the correct distinction between the final two candidates. (Apocope could be ranked either higher or lower than ClosedFtHd for this exam-

ple to come out correctly.)

Adding Apocope also corrects the problem for (77b) as long as it is ranked higher than ClosedFtHd as demonstrated in (81) and (82).

(81)

Candidates	ParseC	Apocope	ClosedFtHd	ParseV
$[baa]_F[ba.nV]_F$		*!	**	
☞ $[baa]_F[ban]_F$Ⓥ			*	*
$[baab]_F$ⓐ$[nV]_\sigma$		*!		*

The correct form is selected in (81). Apocope rules out the unattested $[baab]_F[nV]_\sigma$. Ranking ClosedFtHd over Apocope fails to rectify the inaccuracy, as (82) demonstrates.

(82)

Candidates	ParseC	ClosedFtHd	Apocope	ParseV
$[baa]_F[ba.nV]_F$		**!	*	
$[baa]_F[ban]_F$Ⓥ		*!	*	*
☞ $[baab]_F$ⓐ$[nV]_\sigma$			*	*

As we noted above, Apocope is just a special case of ClosedFtHd. It is no surprise, then, that Apocope should be ranked higher than ClosedFtHd.[19]

Since Apocope is a special case of ClosedFtHd, Apocope appears to be redundant. It may well be possible, however, to account for the data employing different constraints which avoid such redundancy. One such conceivable constraint would be as in (83)

(83) AvoidLightSyl: $*\breve{\sigma}$

[19]This is handled by Pāṇini's Theorem on Constraint-ranking (Prince & Smolensky 1993b:82) in Optimality Theory.

That is, light syllables are to be avoided wherever possible. This constraint would necessarily interact with syllabification constraints and would often be violated in attested forms.

While it would reflect the basic process in syncope (i.e. the removal of a light syllable), it seems counter-intuitive at first glance to posit that the most highly attested syllable shape (cf. Clements & Keyser 1983:29 and Prince & Smolensky 1993b:85) should be precisely the shape to avoid. Optimality Theory does posit other seemingly counter-intuitive constraints such as *Struc: avoid structure (McCarthy & Prince 1993:22). As mentioned in section 1.3.2, the constraint *Struc, like AvoidLightSyl, does not mean that structure is bad. Rather it serves simply to minimize the amount of structure in the optimal candidate. This type of a constraint is quite distinct from the type of constraints we have been considering. Rather than constituting a statement about the well-formedness of a representation, such constraints serve to control the proliferation of candidate sets produced by the generator function. These constraints are quite intuitive when considered in terms of their "minimize" function.[20]

Unfortunately, there is an empirical difficulty with using the AvoidLightSyl constraint. Replacing Apocope with AvoidLightSyl will not always select the correct candidate. For example, the tableau for ku.sup in (84) shows that the addition of AvoidLightSyl erroneously selects both of the final two candidates as optimal.

(84)

Candidates	ParseC	AvoidLightSyl	ClosedFtHd	ParseV
[ku.su]$_F$[pa]$_\sigma$		***!	*	
☞[kus]$_F$ⓤ[pa]$_\sigma$		*		*
☞[ku.sup]$_F$ⓐ		*		*

[20] As also noted in section 1.3.2, Constraint-Ranked Derivation has no need for these minimizing constraints.

The first candidate has more violations of AvoidLightSyl than the other two candidates and is therefore eliminated. The other two candidates have an equal number of violations of both AvoidLightSyl and ParseV. Only the third should be selected as optimal.

Another conceivable constraint has been suggested by Junko Itô (p.c.). It would be one that insists that every syllable be parsed into a foot (it is a part of the family of Parse constraints which correspond to "Prosodic Licensing" Itô 1986, Goldsmith 1990, Itô & Mester 1993). (Alternatively, it could be a constraint asserting that unfooted syllables are to be avoided.) As a member of the Parse family of constraints, it is independently required.

(85) Parse-σ: $*$ F
ǂ
σ

If Parse-σ were ranked above ClosedFtHd, it would make the correct selection in the problematic cases of (78) and especially (79). This is evidenced by the tableaux in (86) and (87), respectively.

(86)

Candidates	ParseC	Parse-σ	ClosedFtHd	ParseV
$[\text{ku.su}]_F[\text{pa}]_\sigma$		*!	*	
$[\text{kus}]_F$ⓤ$[\text{pa}]_\sigma$		*!		*
☞ $[\text{ku.sup}]_F$ⓐ				*

(87)

Candidates	ParseC	Parse-σ	ClosedFtHd	ParseV
$[\text{baa}]_F[\text{ba.nV}]_F$			**!	
☞ $[\text{baa}]_F[\text{ban}]_F$Ⓥ			*	*
$[\text{baab}]_F$ⓐ$[\text{nV}]_\sigma$		*!		*

Thus, both the parallel candidate set approach and the Constraint-Ranked Derivation approach are able to provide the same insights in analyzing this set of data.

2.3.4 Stem Binarity

While ParseC, ClosedFtHd, and ParseV provide an explanation for the data seen so far, these are not enough to explain all of the Southeastern Tepehuan forms. Consider the data in (88).[21]

(88)	a.	/sui.sui.ma.ri/	⟶	[suis]$_F$[mar̃]$_F$	'deer (pl)'	(=34a)
	b.	/hin-nuu.nuu.ti.sV/	⟶	hiñ-[ñuun]$_F$[čiš]$_F$	'my brothers-in-law'	(=34b)
	c.	/gio.gio.tɨ.rV/	⟶	[gio'ŋ]$_F$[tɨr]$_F$	'plains'	
	d.	/pii.pii.pi.ri/	⟶	[piip]$_F$[pir̃]$_F$	'chicks'	

Unlike the forms in section 2.3.3, the initial syllabification and footing of the stem for these forms will produce three feet. For example, the prosodic parsing of (88d) is [pii]$_F$[pii]$_F$[pi.ri]$_F$. The best that the ClosedFtHd constraint and repair can do is to close the head of the final foot: [pii]$_F$[pii]$_F$[pir]$_F$. Something else must reduce it to the surface shape.

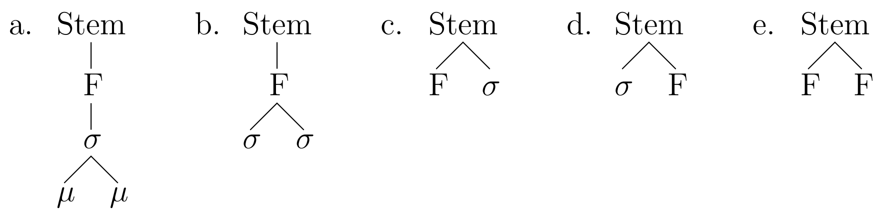
What these forms do have in common with the previous data is that the surface form of the stem is always binary. That is, the prosodic configuration always involves binary branching. This is reminiscent of the situation noticed for many Japanese templatic formations by Itô & Mester (1992:21). I modify their Word Binarity structure requirement as follows:

(89) StemBin: Stems must be prosodically binary.

[21]The second s in (88a) would be expected to undergo palatalization, but apparently does not. It appears to be exceptional since palatalization occurs in other instances of the ui diphthong such as in duduiñkar 'pipes'.

In (88d), two identical consonants surface side-by-side. This is to be contrasted with Tonkawa which prohibits such configurations via the OCP (McCarthy 1986:223–225).

I follow their concept of Weak-Layering (whereby prosodic structures must be properly headed and maximally parsed, but need not obey "strict layering") which allows for all of the prosodic structures in (90) to meet Stem Binarity.[22]

(90)


This constraint captures the observation that practically all of the Southeastern Tepehuan forms have stems which involve such binary configurations.[23]

Independent support for Stem Binarity can be gleaned from the compound forms given in E. Willett (1981b:33). E. Willett notes that when two longer nouns are compounded, the compound contains two "accent centers." That is, the compound consists of two words (91a). Shorter nouns result in a compound with one "accent center" and constitute a single word (91b).

(91)

	Compound	Prosodic Bracketing	Gloss
a.	kɨi'ñ.kam ko'	[kɨi'ñ]$_F$[kam]$_F$ [ko']$_F$	'rattlesnake'
b.	bi'ñ.vak	[bi'ñ]$_F$[vak]$_F$	'clay house'

In our terms, this means that a compounded noun can be treated as a single word only if the entire compound meets Stem Binarity. Otherwise, it must be treated as two words, each of which will independently meet Stem Binarity.[24]

[22]Itô & Mester (1992) do not allow for structures such as (90a). Since Southeastern Tepehuan clearly requires such structures (see (43c–d,f)), we can account for this by positing a parametric distinction between the Southeastern Tepehuan and Japanese cases.

[23]Note that the exceptional forms to apocope, (46) voohi 'bear' and its plural vapoohi, do meet Stem Binarity. They have the pattern of (90c).

[24]There is probably more at play here since the word for 'house' independently occurs as va'aak. Why it is shortened in this compound is not clear.

2.3.4.1 Stem Binarity Constraint

We will now address the question of how Stem Binarity is enforced. One possibility might be for it to be a template to which the stem is mapped (McCarthy & Prince 1986, 1988, 1990). Another is for it to be a well-formedness condition with a repair strategy which is triggered whenever the condition is violated (Paradis 1988).

Suppose that Stem Binarity is a template to which the stem is mapped. Besides the fact that there is not a specific template (Stem Binarity is a configuration with various categorial realizations; see Itô & Mester 1992:16), such a mapping approach would have difficulty accounting for the forms in (92).

(92)	a.	/huutu-d/	⟶	[huu]$_F$[tu'n]$_F$	'his fingernail'	(=43a.i)
	b.	/tuu.tu.vuu.ri-d/	⟶	[tuut]$_F$[vu.r̃i'ñ]$_F$	'his hips'	(=33d)
	c.	/vaa.poo.tV.po.da.gV/	⟶	[va.poot]$_F$[po.da']$_F$	'worms'	(=62b)

Suppose the template were filled from right-to-left. Underlyingly, both (92a) and (92b) end in a [cvv][cvc] sequence. In (92a), the initial long syllable remains long (and an independent foot). In (92b), however, the corresponding long syllable must shorten. If the template were to be filled from left-to-right, on the other hand, the initial long vowel in (92b) must remain long, while the initial long vowel in (92c) must shorten. It is not clear how such a distinction could be made in a template filling process.

The other way to view the Stem Binarity enforcement is as a constraint which has an associated repair strategy to fix-up the violation. Such a constraint is stated in (93).

(93) StemBin: Stems should be binary at some level of prosody. (=89)

As in the case for ClosedFtHd, the designated repair strategy for Southeastern Tepehuan is remove-μ and the repair sites are determined from the right to the left.

Given that both ClosedFtHd and StemBin involve the same repair strategy and a similar mechanism for determining the locus of repair, are they really independently needed? That the answer is in the affirmative is demonstrated by the examples in (94).

(94)	a.	/na.ka.sɨ.rV/	⟶	[nak]$_F$[sɨr]$_F$	'scorpion'	(=34c)
	b.	/tuu.tu.vuu.ri-d/	⟶	[tuut]$_F$[vu.r̃i'ñ]$_F$	'his hips'	(=33d)

If StemBin were the only constraint, then we would expect (94a) to surface as *[na.ka]$_F$[sɨ.rV]$_F$. This output satisfies StemBin because it consists of two feet, but neither foot is headed by a closed syllable. On the other hand, if ClosedFtHd were the only constraint, then (94b) would be the tri-footed *[tuut]$_F$[vuu]$_F$[r̃i'ñ]$_F$. The repair for ClosedFtHd could not do any better than this.

Since ClosedFtHd and StemBin are thus independently required constraints, they must be ranked with respect to each other. The forms in (95) provide the crucial data for determining the ranking.

(95)	a.	/pii.pii.pi.ri/	⟶	[piip]$_F$[pir̃]$_F$	'chicks'	(=88d)
	b.	/sui.sui.ma.ri/	⟶	[suis]$_F$[mar̃]$_F$	'deer (pl)'	(=34a)

Suppose that ClosedFtHd is ranked above StemBin. The derivation for (95a) would proceed as in (96).

(96)

	Representation	Constraint	Repair	OK?
Input	[pii]$_F$[pii]$_F$[pi.ri]$_F$	ParseC		
a.		*ClosedFtHd	↦ [pii]$_F$[pii]$_F$[pir]$_F$i	√
b.	[pii]$_F$[pii]$_F$[pir]$_F$i		↦ [pii]$_F$[piip]$_F$iri	*
c.			↦ [pii]$_F$[pi.pir]$_F$i	*
		*StemBin	[pii]$_F$[pi.pir]$_F$i	√
	[pii]$_F$[pi.pir]$_F$i	*ParseV	N/A	
Output	[pii]$_F$[pi.pir]$_F$i			
Surface	*[pii]$_F$[pi.pir̃]$_F$			

All three feet would violate ClosedFtHd. The repairs would be applied iteratively from right-to-left, once for each foot in turn. The first repair is shown in line (a). It would successfully close the final foot ([pi.ri]$_F$) and would thereby become the representation of choice. Line (b) shows the repair for the second foot ([pii]$_F$), which would produce a violation of ParseC. The attempted repair for the initial foot ([pii]$_F$) would also be unsuccessful as shown in line (c). Since it would fail to close the initial foot, it would be a violation of ClosedFtHd itself.[25] Per the Constraint-Ranked Derivation algorithm discussed in chapter 1, this initial foot (the domain of attempted repair) would be marked as exceptional to the ClosedFtHd constraint. This prevents lower-ranked constraints from failing due to the failure of a higher-ranked constraint's ability to repair the violation of a particular domain. StemBin, of course, would be violated and repaired as shown. The initial foot [pii]$_F$ would not meet ClosedFtHd, but it is marked as exceptional to ClosedFtHd. Nonetheless, the resulting representation is incorrect.

If StemBin is ranked above ClosedFtHd, however, the attested form is produced as illustrated in (97).

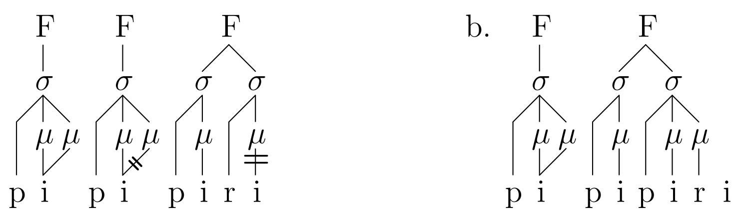
[25] Only one invocation of the repair strategy is allowed per syllable. Therefore, a double application of remove-μ to the medial syllable in line (c) is not possible even though it would cause the initial foot to become closed: [piip]$_F$[pir].

(97)

	Representation	Constraint	Repair	OK?
Input	[pii]$_F$[pii]$_F$[pi.ri]$_F$	PARSEC		
		∗STEMBIN	[pii]$_F$[pi.pir]$_F$i	√
	[pii]$_F$[pi.pir]$_F$i	∗CLOSEDFTHD	↦ [piip]$_F$[pir]$_F$i	√
	[piip]$_F$[pir]$_F$i	∗PARSEV	N/A	
Output	[piip]$_F$[pir]$_F$i			
Surface	[piip]$_F$[pir̃]$_F$			

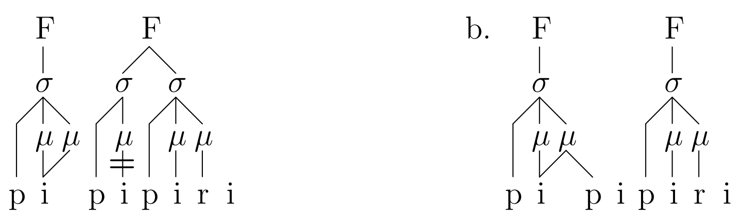
The crucial step is the repair for STEMBIN. The original three feet must be reduced to two. The repair site is determined in a right-to-left sweep, resulting in the final two feet being targeted for reduction. By removing the word-final mora and shortening the long vowel in the middle foot, binarity is obtained as illustrated in (98).

(98) a. b.



The initial foot head of the resulting representation is not closed and so violates CLOSEDFTHD. Its associated repair produces the attested form as shown in (99).

(99) a. b.



Note that the serial application of the two constraints and their associated repairs leads to the removal of a long vowel. Such a reduction is otherwise unattested as a syncope pattern. For example, in Tonkawa, a long vowel is shortened, not removed (Kenstowicz & Kisseberth 1979:431). This also suggests that there must be two constraints, not merely one which would apply repeatedly.

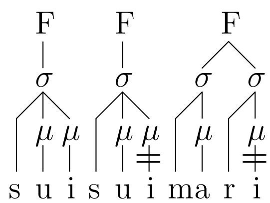
Diphthongs can also reduce as the derivation in (100) for (95b) illustrates.

(100)

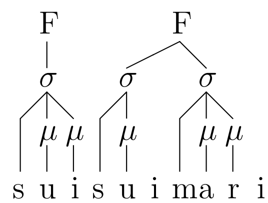
	Representation	Constraint	Repair	OK?
Input	$[\text{sui}]_F[\text{sui}]_F[\text{ma.ri}]_F$	ParseC		
		*StemBin	$[\text{sui}]_F[[\text{su}]_\sigma\text{i}[\text{mar}]_\sigma]_F\text{i}$	√
	$[\text{sui}]_F[[\text{su}]_\sigma\text{i}[\text{mar}]_\sigma]_F\text{i}$	*ClosedFtHd	↦ $[\text{suis}]_F\text{ui}[\text{mar}]_F\text{i}$	√
	$[\text{suis}]_F\text{ui}[\text{mar}]_F\text{i}$	*ParseV	N/A	
Output	$[\text{suis}]_F\text{ui}[\text{mar}]_F\text{i}$			
Surface	$[\text{suis}]_F[\text{mar̃}]_F$			

This parallels the derivation for (95a). The result of the violation and repair for StemBin is given in (101).

(101) a.

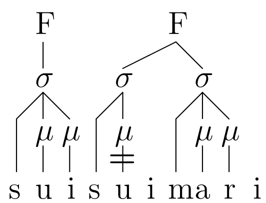


b.

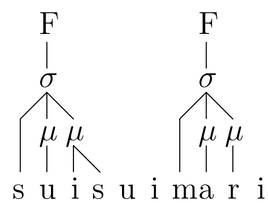


The diagram in (102) shows the results of the repair for ClosedFtHd.

(102) a.



b.

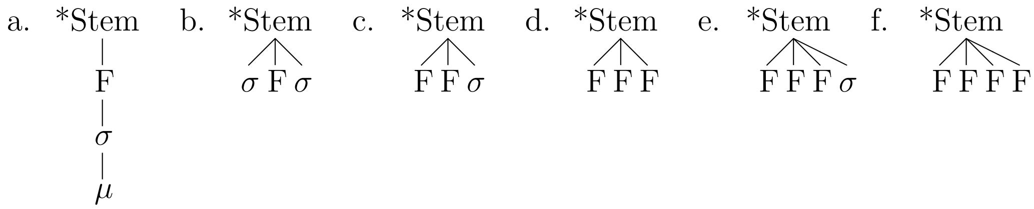


The ranking of constraints, then, becomes the one in (103).

(103) $<\text{StemFormation} : H> (R_{In}) \ = \ R_{Out},$
where H = ParseC ≫ StemBin ≫ ClosedFtHd ≫ ParseV

2.3.4.2 Binarity Exceptions

Not every single Southeastern Tepehuan nominal form meets Stem Binarity. It will be useful to consider those configurations that do not meet Stem Binarity:

(104)


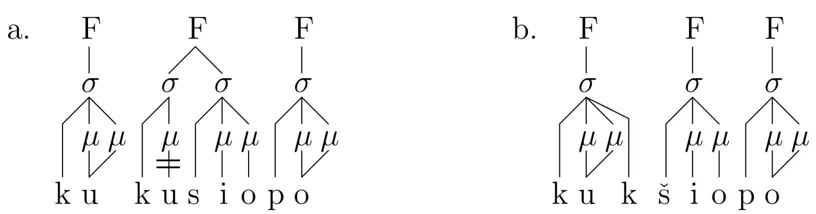
The first is mono-moraic. T. Willett (1991:26–27) notes that there are no mono-moraic stems in Southeastern Tepehuan. Since there are mono-syllabic, bi-moraic forms (such as gaa 'cornfield'), this implies that the minimal word in Southeastern Tepehuan is bi-moraic.

The ternary configuration in (104b) is also unattested. This is to be expected given that footing is from left-to-right.[26] Configurations (104c–f) are all possible and do arguably occur after initial prosodic parsing of underlying stems. All but a few of these reduce to a binary configuration at surface level.

Those few all have the configuration in (104d). These tri-footed words are shown in (105).[27]

(105)	a.	/ha-kuu.ku.sio.poo/	⟶	ha-[kuuk]$_F$[šio]$_F$[poo]$_F$	'their necks'
	b.	/koo.kV.roi.da.gV/	⟶	[kook]$_F$[roi]$_F$[ǰa']$_F$	'tadpole'
	c.	/koo.koo.kV.roi.da.gV/	⟶	[ka.kook]$_F$[roi]$_F$[ǰa']$_F$	'tadpoles'

The stem portion of (105a) is illustrated in (106).

(106)


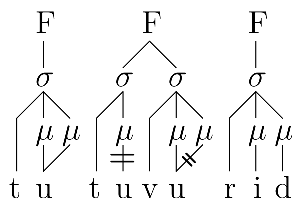
Compare these with the similar forms in (107), which do meet Stem Binarity.

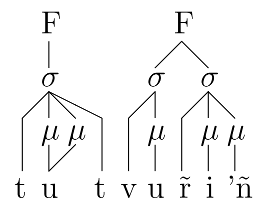
[26] Similarly, (90d) does not arise since no skipping is allowed in foot parsing.

[27] These three forms are the only exceptions to Stem Binarity I have found. The identification of the final glottal as an underlying g in (105b–c) and in (107b) is a conjecture. A dissimilation process causes an o to become an a when its syllable is not a foot head as in (105c).

(107) a. /tuu.tu.vuu.ri-d/ ⟶ [tuut]$_F$[vu.r̃i'ñ]$_F$ 'his hips' (=33d)
b. /taa.ta.kaa.rui.gV/ ⟶ [taat]$_F$[ka.rui']$_F$ 'chickens'

In (107a), the penultimate syllable [vuu]$_\sigma$ is shortened to [vu]$_\sigma$, thereby forming a canonical iamb with the following syllable. This is illustrated in (108) (along with the closing of the initial foot head).

(108) a.


b.


Notice how the shortening process applies to create an output that is binary at the Stem level.

The primary difference between the sets of forms in (105) and (107) lies in the surface penultimate syllable. Underlyingly, what becomes the penult in (105) contains a diphthong while in (107) it has a long vowel. Similarly, the syllable which surfaces as the penult in (107) is short, while in (105) it remains a diphthong.

The data to be accounted for, then, are exemplified by the two forms repeated in (109).

(109) a. /ha-kuu.ku.sio.poo/ ⟶ ha-[kuuk]$_F$[šio]$_F$[poo]$_F$ 'their necks' (=105a)
b. /tuu.tu.vuu.ri-d/ ⟶ [tuut]$_F$[vu.r̃i'ñ]$_F$ 'his hips' (=33d)

Only ClosedFtHd is at work in (109a); Stem Binarity is not met. By way of contrast, (109b) not only closes the heads of the initial and final feet, but the long vowel of the penultimate syllable is shortened. The output meets binarity (as well as exceptionlessly meeting ClosedFtHd).

The derivation for (109b) is as in (110).

(110)

	Representation	Constraint	Repair	OK?
Input	$[\text{tuu}]_F[\text{tu.vuu}]_F[\text{ri-d}]_F$	PARSEC		
		*STEMBIN	$[\text{tuut}]_F\text{u}[\text{vu.ri-d}]_F$	√
	$[\text{tuut}]_F\text{u}[\text{vu.ri-d}]_F$	CLOSEDFTHD		
		*PARSEV	N/A	
Output	$[\text{tuut}]_F\text{u}[\text{vu.ri-d}]_F$			
Surface	$[\text{tuut}]_F[\text{vu.ri-d}]_F$			

The repair for STEMBIN targets the removal of the middle foot. The shortening of vuu and the removal of the mora in tu produces the required binary representation.

The tri-footed form in (109a) differs in the presence of a diphthong in the head of the second foot. Its derivation would be predicted to be as in (111) (only the stem portion is shown).

(111)

	Representation	Constraint	Repair	OK?
Input	$[\text{kuu}]_F[\text{ku.sio}]_F[\text{poo}]_F$	PARSEC		
		*STEMBIN	$\mapsto [\text{kuuk}]_F\text{u}[[\text{si}]_\sigma\text{o}[\text{poo}]_\sigma]_F$	√
	$[\text{kuuk}]_F\text{u}[[\text{si}]_\sigma\text{o}[\text{poo}]_\sigma]_F$	*CLOSEDFTHD	$\mapsto [\text{kuuk}]_F\text{u}[[\text{si}]_\sigma\text{o}[\text{po}]_\sigma]_F$	*
		*PARSEV	N/A	
Output	$[\text{kuuk}]_F\text{u}[[\text{si}]_\sigma\text{o}[\text{poo}]_\sigma]_F$			
Surface	$*[\text{kuuk}]_F[\text{ši.poo}]_F$			

The representation in the CLOSEDFTHD line violates CLOSEDFTHD: the head of the final foot is not closed. The REMOVE-μ repair, however, does not improve the final foot as far as the CLOSEDFTHD constraint is concerned and, therefore, fails to apply. The crucial thing to note here is that the output is incorrect.

Why the medial diphthong in a form like (95b) sui.sui.ma.ri is removed whereas the diphthong in (109a) ha-kuu.ku.sio.poo remains intact is unclear (see also (105)). For now, I will posit that the exceptional forms like (109a) are lexically marked as

exceptions to StemBin.[28] That is, the StemBin constraint is made inactive for (109a). The derivation will then proceed as in (112).

(112)

	Representation	Constraint	Repair	OK?
Input	[kuu]$_{F}$[ku.sio]$_{F}$[poo]$_{F}$	ParseC		
		(StemBin)		
		∗ClosedFtHd	↦ [kuu]$_{F}$[ku.sio]$_{F}$[po]$_{F}$	∗
			↦ [kuu]$_{F}$[ku.si]$_{F}$o[poo]$_{F}$	∗
			↦ [kuuk]$_{F}$u[sio]$_{F}$[poo]$_{F}$	√
	[kuuk]$_{F}$u[sio]$_{F}$[poo]$_{F}$	∗ParseV	N/A	
Output	[kuuk]$_{F}$u[sio]$_{F}$[poo]$_{F}$			
Surface	[kuuk]$_{F}$[šio]$_{F}$[poo]$_{F}$			

The parentheses around StemBin indicate that it is not applicable per the lexical exception. There are three violations of ClosedFtHd. The repairs for the first two fail to improve the targeted foot and so are not employed. The third repair succeeds in resolving the violation.

2.3.4.3 Sample Derivations

Several more sample derivations will now be given to illustrate the role of the repair of StemBin. Derivations are provided for the forms in (113).

(113)

a.	/naa.na.ka.sɨ.rV/	⟶	[naan]$_{F}$[ka.sɨr]$_{F}$	'scorpions'	(=62a)
b.	/taa.ta.kaa.rui.gV/	⟶	[taat]$_{F}$[ka.rui']$_{F}$	'chickens'	(=107b)
c.	/haa.haa.vV.ka.ri-d/	⟶	[ha.haav]$_{F}$[ka.r̃i'ñ]$_{F}$	'his lungs'	

The derivation in (114) is for (113a).

[28] One difference between the two cases is that when the diphthong is removed, its melody is copied in the reduplicative template. There may, then, be some kind of abstract recoverability at issue. This recoverability would need to apply uniquely to diphthongs since other vowels are obviously deleted quite freely in Southeastern Tepehuan.

(114)

	Representation	Constraint	Repair	OK?
Input	$[\text{naa}]_F[\text{na.ka}]_F[\text{sɨ.rV}]_F$	PARSEC		
		*STEMBIN	↦ $[\text{naan}]_F\text{a}[\text{ka.sɨr}]_F\text{V}$	√
	$[\text{naan}]_F\text{a}[\text{ka.sɨr}]_F\text{V}$	CLOSEDFTHD		
		*PARSEV	N/A	
Output	$[\text{naan}]_F\text{a}[\text{ka.sɨr}]_F\text{V}$			
Surface	$[\text{naan}]_F[\text{ka.sɨr}]_F$			

This is another case where the repair for STEMBIN reduces the initial three feet to two. The word-final mora and the initial mora of the middle foot are targeted for REMOVE-μ. This causes the middle foot to merge with the outer feet. The result of the merger happens to also close the heads of the remaining feet.

The next derivation is for (113b), a case whose underlying form has three feet plus a light syllable ($\breve{\sigma}$). (This is Stem Binarity case (104e).)

(115)

	Representation	Constraint	Repair	OK?
Input	$[\text{taa}]_F[\text{ta.kaa}]_F[\text{rui}]_F[\text{gV}]_\sigma$	PARSEC		
		*STEMBIN	↦ $[\text{taat}]_F\text{a}[\text{ka.ruig}]_F\text{V}$	√
	$[\text{taat}]_F\text{a}[\text{ka.ruig}]_F\text{V}$	CLOSEDFTHD		
		*PARSEV	N/A	
Output	$[\text{taat}]_F\text{a}[\text{ka.ruig}]_F\text{V}$			
Surface	$[\text{taat}]_F[\text{ka.rui'}]_F$			

The three-plus feet are reduced to two in a right-to-left sweep. The final foot and light syllable are merged to produce a mono-syllabic heavy. This is then merged with the preceding foot to form a canonical iamb. The head of the $[\text{ta.kaa}]_F$ foot is targeted for REMOVE-μ and is incorporated into the following foot. The non-head of the $[\text{ta.kaa}]_F$ foot must also lose its mora in order to attain binarity. The t̠ onset becomes a coda.

The final derivation is for (113c). Initial footing produces four feet (which is Stem Binarity case (104f)).

(116)

	Representation	Constraint	Repair	OK?
Input	$[\text{haa}]_F[\text{haa}]_F[\text{v}\textsc{v}.\text{ka}]_F[\text{rid}]_F$	ParseC		
		∗StemBin	↦ $[\text{ha.haav}]_F\textsc{v}[\text{ka.rid}]_F$	√
	$[\text{ha.haav}]_F\textsc{v}[\text{ka.rid}]_F$	ClosedFtHd		
		∗ParseV	N/A	
Output	$[\text{ha.haav}]_F\textsc{v}[\text{ka.rid}]_F$			
Surface	$[\text{ha.haav}]_F[\text{ka.ri'ñ}]_F$			

Once again, the repair for StemBin does all the work. To merge the final two feet, the mora of the non-head is removed. This produces a final canonical iamb and closes the head of the second foot. The first and second feet are merged into one by shortening the head of the initial foot.[29]

2.3.4.4 Stem Binarity and Parallel Candidate Sets

Having seen how ClosedFtHd and StemBin under the Constraint-Ranked Derivation approach to optimization accounts for the Southeastern Tepehuan nominal forms, how does the parallel candidate set approach fare? Section 2.3.3.2 discussed how the candidate set approach accounted for the data as long as an additional constraint besides ClosedFtHd was posited. Since a constraint such as Parse-σ would be independently required, this does not pose any problem for that approach. This section will demonstrate that an additional constraint is also needed to account for the Stem Binarity issues under the parallel candidate set approach. It, too, is arguably independently required.

[29] As can be seen from the derivations in (114)–(116), a foot head may become closed in order to meet StemBin. In fact, a form such as (95a) $[\text{pii}]_F[\text{pii}]_F[\text{pi.ri}]_F$ could meet StemBin by reducing to $[\text{pi.pii}]_F[\text{pi.ri}]_F$ (although this would be to apply the repairs in a left-to-right rather than a right-to-left direction). That it reduces to $[\text{pii}]_F[\text{pi.pir}]_F$ and then to $[\text{piip}]_F[\text{pir}]_F$ implies that the closed foot head target is somehow in view. Thus, there appears to be some kind of interrelatedness between these two independently required constraints. (Recall also that Apocope is subsumed by ClosedFtHd). This suggests that there may be some less obvious overriding factor that drives the observed surface configurations. What that might be is not clear to me and remains a matter for future research.

Recall that in order to meet StemBin, occasionally an underlying long syllable will be shortened. Two such cases are repeated in (117).

(117)	a.	/tuu.tu.vuu.ri-d/	⟶	[tuut]$_F$[vu.r̃i'ñ]$_F$	'his hips'	(=33d)
	b.	/haa.haa.vV.ka.ri-d/	⟶	[ha.haav]$_F$[ka.r̃i'ñ]$_F$	'his lungs'	(=113c)

The candidate set approach must produce the appropriate surface forms. This entails that for every underlying long vowel, two candidates must be produced: one with a long vowel and one with a shortened vowel. For example, the tableau in (119) shows three relevant candidates for (117a). The _ symbol indicates a delinked mora as illustrated in (118).

(118)

u_

StemBin is ranked above ClosedFtHd as in the Constraint-Ranked Derivation approach. (Recall from section 2.3.3.2 that Parse-σ is above ClosedFtHd; I have arbitrarily ranked StemBin above Parse-σ.) The constraint names have been abbreviated in order to fit the tableau on the page.[30]

(119)

Candidates	ParseC	SB	Par-σ	CFH	ParseV
[tuut]$_F$ⓤ[vuu]$_F$[r̃i'ñ]$_F$		*!		*	*
☞[tu_t]$_F$ⓤ[vu_.r̃i'ñ]$_F$					*
☞[tuut]$_F$ⓤ[vu_.r̃i'ñ]$_F$					*

The underlying vuu remains long in the first candidate. The result, of course, violates StemBin. This syllable is shortened in both of the other two candidates. The second candidate also has the initial tuu syllable shortened while the third candidate does

[30]SB is StemBin, CFH is ClosedFtHd, and Par-σ is Parse-σ. Recall that an encircled letter indicates an unparsed segment.

not. The set of constraints is not able to distinguish between the second and third candidates: both satisfy all constraints except PARSEV, which they equally violate.

The difference between the two selected candidates is the length of the initial closed syllable. Since it is closed, one might suspect that length is not really significant; both [tut]$_F$ and [tuut]$_F$ will be bi-moraic. That every underlying long vowel must have candidates both with and without length, however, predicts that a form such as the one in (120) will have the candidates shown in the tableau of (121).

(120) /baa.banV/ ⟶ [baa]$_F$[ban]$_F$ 'coyotes' (=71c)

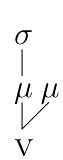
(121)

Candidates	PARSEC	SB	PAR-σ	CFH	PARSEV
[baa]$_F$[ba.nV]$_F$			*!	**	
☞[ba_.ban]$_F$Ⓥ					*
[baa]$_F$[ban]Ⓥ				*!	*

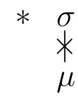
The final two candidates are clearly prosodically distinct in this case. The first would bear stress on the ultima, ban; the second would have stress on the penult, baa. Unfortunately, the incorrect candidate is chosen.

It appears, then, that the candidate set approach must posit yet another constraint to handle the Southeastern Tepehuan data. Prince & Smolensky (1993b:60) discuss a similar situation for Latin and treat such shortened vowels as having a syllabically underparsed mora. The representation is as in (122).

(122) Syllabically Unparsed Mora (Prince & Smolensky 1993b:60)



Such a configuration constitutes a violation of PARSE-μ.

(123) Parse-μ: 

The constraint insists that every mora be parsed into a syllable.

At first glance, this appears to solve the problem. We can add Parse-μ to the hierarchy, ranking it just above Parse-σ, and the correct candidate will be selected for the case of (120). The appropriate tableau is given in (124). (The highest ranked ParseC is omitted for formatting purposes and unparsed moras are indicated by an underlined vowel.)

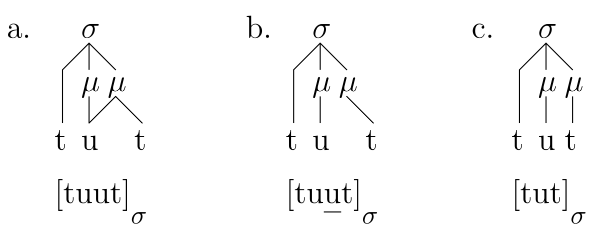
(124)

Candidates	SB	Parse-μ	Par-σ	CFH	ParseV
$[\text{baa}]_F[\text{ba.nV}]_F$			*!	**	
$[\text{ba}\underline{\text{a}}\text{.ban}]_F$Ⓥ		*!			*
☞ $[\text{baa}]_F[\text{ban}]$Ⓥ				*	*

The second candidate is eliminated because it violates Parse-μ. The first candidate has two violations of ClosedFtHd while the third has but one. The third is therefore selected as the best candidate.

Unfortunately, however, Parse-μ does not help for the $[\text{tuut}]_F[\text{vu.r̃i'ñ}]_F$ case of (117a). The initial foot $[\text{tuut}]_F$ must be bi-moraic. Therefore, both moras must be parsed. Each of the final two candidates will have one violation of Parse-μ and the constraint ranking will not be able to distinguish the candidates.

The distinction between the two crucial candidates is whether or not the final consonant is moraic as illustrated in (125).

(125) 

There is a clear representational distinction between (125a) (long vowel, closed syllable) and (125c) (short vowel, closed syllable). The representation is not able, however, to distinguish between (125b) (a shortened long vowel, closed syllable) and (125c) (short vowel, closed syllable). PARSE-μ, then, cannot be appealed to in order to solve this problem.

A possible constraint that will be able to distinguish the two candidates would be one that relates the underlying length to the surface length. This is, after all, the crucial distinction between the attested forms and the incorrectly selected ones. Such a constraint is given in (126).

(126) UNDERLENGTH: Underlying length should be preserved.

That is, underlying length is expected to surface. This means that every 'delinking' of an underlying 'link' counts as a violation of UNDERLENGTH.

We can rank this constraint just above PARSE-σ and obtain the correct results as the tableaux in (127) and (128) show.[31]

(127)

Candidates	PARC	SB	UL	PAR-σ	CFH	PARV
[baa]$_F$[ba.nV]$_F$				*!	**	
[ba_.ban]$_F$Ⓥ			*!			*
☞[baa]$_F$[ban]Ⓥ					*	*

(128)

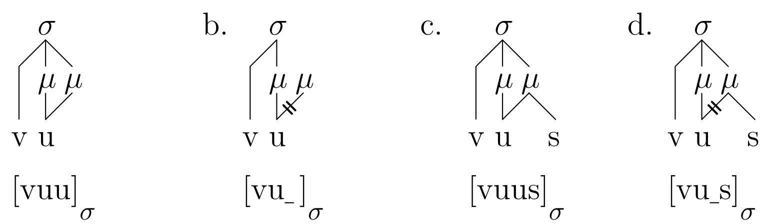
Candidates	PARC	SB	UL	PAR-σ	CFH	PARV
[tuut]$_F$ⓤ[vuu]$_F$[r̃i'ñ]$_F$		*!			*	*
[tu_t]$_F$ⓤ[vu_.r̃i'ñ]$_F$			**!			*
☞[tuut]$_F$ⓤ[vu_.r̃i'ñ]$_F$			*			*

[31]More abbreviations are employed to fit the tableaux on the page. PARC is PARSEC, UL is UNDERLENGTH, and PARV is PARSEV.

The UnderLength constraint has a different characteristic from all of the other constraints we have seen. The others are all statements about the surface form: consonants/vowels should be parsed; the stem should be prosodically binary; heads of feet should be closed; and final light syllables should be avoided. UnderLength, on the other hand, is a statement about the relationship between the moraic structure of a root node in the lexicon and its moraic structure at the surface level. This is reminiscent of the Two-Level approach to phonology of Koskenniemi (1983) (other work in this theory includes Karttunen 1983, 1991, Karttunen, Koskenniemi, & Kaplan 1987, and Antworth 1990). This theory posits that there is a one-to-one correspondence between surface and lexical segments. The phonological rules have simultaneous access to both the underlying and surface form of each and every segment. Obviously, such an approach requires additional machinery to match up underlying and surface segments.

Another possible way to allow for the information needed by a constraint such as UnderLength is to allow candidates to have syllables represented as in (129).[32]

(129) a. b. c. d.



The representation for an underlying long vowel that is realized as long would be as in (129a). Example (129b) would represent a shortened underlying long vowel. Similarly, (129c–d) represent underlying long vowels with a coda consonant included in the surface syllable.

[32] McCarthy & Prince (1993:21) assume that every mora can dominate at most one root node. Thus the representation in (129c) would need to have the final consonant attach to the syllable node. I have chosen to represent all codas as moraic to (i) provide consistency in their representation and (ii) to allow for a moraically oriented means of distinguishing onsets from codas. Onsets are consonants which attach directly to the syllable node; codas are consonants which attach to a mora.

The delinked line would be a crucial part of the representation. It would denote that while the underlying form has the indicated constituency, the candidate does not. A constraint such as UNDERLENGTH would have access to this representation and every instance of a delinked constituency line would constitute a violation of the constraint.

The need for an additional constraint such as UNDERLENGTH is a direct consequence of the parallel candidate set approach. Where a phonology which allows sequential application of processes does not require surface forms to maintain a record of their underlying form, the candidate set approach must allow for such a record-keeping mechanism at least for certain configurations. This implies that a constraint like UNDERLENGTH is independently required under this approach.[33]

2.3.4.5 Summary of Constraints

Before directly addressing reduplication in Southeastern Tepehuan, a brief summary of the constraints posited in the analysis is provided. The table in (130) delineates the constraints.

(130)

Constraint	Description	Employs Repair Strategy?
PARSEC	Consonants are parsed	No
STEMBIN	Stems are prosodically binary	Yes
CLOSEDFTHD	Foot heads are closed	Yes
PARSEV	Vowels are parsed	No

The designated repair strategy for Southeastern Tepehuan is REMOVE-μ. It is applied in a right-to-left direction.[34]

[33]The issue of how the parallel candidate set approach could handle lexical exceptions for forms such as (109a) ha-kuu.ku.sio.poo is addressed in section 2.4.3.2.

[34]There is nothing significant about the relative order of constraints with and without repair strategies in (130). For the Pichis Asheninca stress case in chapter 3, we will see an entirely different pattern.

2.4 Reduplication and Binarity

The preceding section demonstrated the key role of the ParseC, ClosedFtHd, and StemBin constraints in determining the prosodic configuration of nominal stems in Southeastern Tepehuan. Many of the examples in that section involve the plural morpheme which is realized by a reduplicative prefix. This section will take a closer look at nominal stem reduplication.

The reduplicated material is always an open syllable, either long or short. The melodic material of the initial (underlying) syllable of the stem is mapped to this open syllable template. E. Willett (1982) and T. Willett (1991) posit that each stem must be lexically marked as to whether it will take the short or long syllable template. It does indeed seem extremely difficult, if not impossible, to find a phonological condition that will account for all of the reduplication forms. I will argue here that a relatively small subset of the forms are lexically marked for reduplication. The rest of the forms will be shown to follow a general pattern of employing a bi-moraic syllable template. This bi-moraic syllable template is shortened to mono-moraic whenever Stem Binarity would be violated.

This section is organized as follows. The data are shown in section 2.4.1 followed by the proposed analysis in section 2.4.2. This subsection discusses the implications of this analysis for the Prosodic Morphology Theory of McCarthy & Prince, especially McCarthy & Prince (1990) and also for the Nuclear Moraic model of Shaw (1992, 1993). The exceptional forms are then presented and discussed in section 2.4.3.

2.4.1 Reduplication Data

Because of the complexity of the truncation patterns, the data are presented in both underlying and surface forms. The data shown in (131) all employ a bi-moraic copy

of the initial syllable of the base. The reduplicated material is underlined.[35]

(131)

	base	singular	plural	glosses
a.	/ba.nV/	ban	baa.ban	‘coyote(s)’
b.	/ku.ra.tV/	ku.rat	kuuk.rat	‘woodpecker(s)’
c.	/na.ka.mɨ.ri/	nak.mɨr̃	naan.ka.mɨr̃	‘bat(s)’
d.	/hin-vu.hi/	hiñ-vui	hiñ-vuu.pui	‘my eye(s)’
e.	/ba.hi-d/	bai’ñ	baa.bai’ñ	‘his tail/their tails’
f.	/koo.gV/	koo’	koo.ko’	‘snake(s)’
g.	/pii.pi.ri/	pii.pir̃	piip.pir̃	‘chick(s)’
h.	/sui.ma.ri/	sui.mar̃	suis.mar̃	‘deer (sg,pl)‘

The prosodic shape of the copy appears to have no necessary relation with the prosodic structure of the base; it is always a bi-moraic syllable even though the initial syllable of the base is mono-moraic in (131a–e) and bi-moraic in (131f–h). The reduplicated form surfaces with two feet in each case.

It appears that only the first syllable of the base is used in mapping the melody to the template. Example (131a) shows that the base is circumscribed before removal of the final vowel. McCarthy & Prince (1990:231) assert that the only prosodic constituent which can be positively circumscribed from the base is the minimal word. This predicts that a form like ban (131a) should reduplicate as the unattested *banban. This problem will be directly addressed in section 2.4.2.1.

By way of contrast, the forms in (132) surface with a mono-moraic copy of the initial syllable of the base.[36]

[35] A process of intervocalic h-drop applies in (131d–e). The final syllable is shortened in (131f); I do not know why.

[36] A hardening process applies in reduplicated forms such as (132b,e,g) causing an underlying fricative v to become the stop p. Dissimilation results in an o becoming an a when its syllable is not a foot head as in (132b). (See also (62b) and (92c)). The g in the base in (132g) is a conjecture.

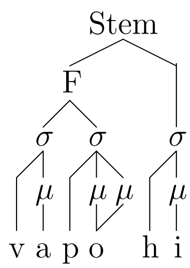
(132)

	base	singular	plural	glosses
a.	/ka.rV.va.si/	kar.vaš	ka.kar.vaš	‘goat(s)’
b.	/voo.hi/	voo.hi	va.poo.hi	‘bear(s)’
c.	/haa.ra.si/	haa.raš	ha.haa.raš	‘crab(s)’
d.	/haa.nV.nu.ri/	haan.nuȓ	ha.haan.nuȓ	‘clothes’
e.	/vai.nu.mV/	vai.ñum	va.pai.ñum	‘metal(s)’
f.	/dui.nV.ka.rV/	duiñ.kar	du.duiñ.kar	‘pipe(s)’
g.	/via.di.kai.gV/	via’ñ.kai’	vi.pia’ñ.kai’	‘lizard(s)’

As in the bi-moraic copy case, the prosodic shape of the base bears no relation to the mono-moraic status of the copy: in (132), the copy is mono-moraic whether the base begins with a mono-moraic syllable (132a) or with a bi-moraic syllable (132b–g).

The surface form of the plural is two feet in every case but (132b).[37] Nevertheless, Stem Binarity is still met in (132b) as (133) shows.

(133)



2.4.2 Binarity and the Reduplication Template

In fact, practically every reduplication form meets Stem Binarity:[38] prosodically, the stem is maximally binary. Every form which employs the bi-moraic copy (131) results

[37]Recall that (132b) is an exception to ClosedFtHd due to the CodaCond constraint which disallows approximants in the coda.

[38]The forms repeated in (i) are exceptions.

(i) a. /ha-kuu.ku.sio.poo/ ⟶ ha-[kuuk]$_F$[šio]$_F$[poo]$_F$ ‘their necks’ (=105a)

b. /ka.koo.kV.roi.da.gV/ ⟶ [koo.kook]$_F$[roi]$_F$[ǰa’]$_F$ ‘tadpoles’ (=105c)

As mentioned in Section 2.3.4.2, their exceptionality may have something to do with the diphthongal nature of the antepenult. They must be lexically marked as exceptional.

in a surface form which is two feet and no more. Every form which employs the mono-moraic copy (132) would create a surface form with more than two feet if a bi-moraic template were used instead. Notice how the mono-moraic copy becomes the weak member of a canonical iambic foot in (132). In all cases but (132b), a bi-moraic copy would produce three feet. In (132b), it would produce two feet plus a syllable. All of these would violate Stem Binarity.

In earlier sections, I have assumed reduplication in Southeastern Tepehuan involves a bi-moraic template. The initial syllable of the base is positively circumscribed and mapped to this template. The representation so produced is then submitted to the Stem Formation process which is subject to the hierarchy of constraints as given in (103).

2.4.2.1 Circumscription of a Syllable

How are we to understand the circumscription and mapping process to this bi-moraic template? McCarthy & Prince (1990:231) restrict the prosodic size of templates to the minimal word:

(134) "Positive prosodic circumscription of a base may only appeal to the category Minimal Word" (McCarthy & Prince 1990:231).

This restriction effectively rules out the unattested case of syllable copy within reduplication systems.

As noted in section 2.3.4.2, the minimal word in Southeastern Tepehuan is bi-moraic: while bi-moraic nominal stems exist, there are no mono-moraic ones (T. Willett 1991:26–27). The Southeastern Tepehuan data, however, appear to require the positive circumscription of the first syllable of the base, even if that syllable is mono-moraic. The need for positive circumscription of a syllable has already been noted by Mester (1990:483-484). His analysis of geisha-house discretionary names in Japanese (which builds on the work of Poser 1990) makes use of a syllable which is

positively circumscribed from the base.

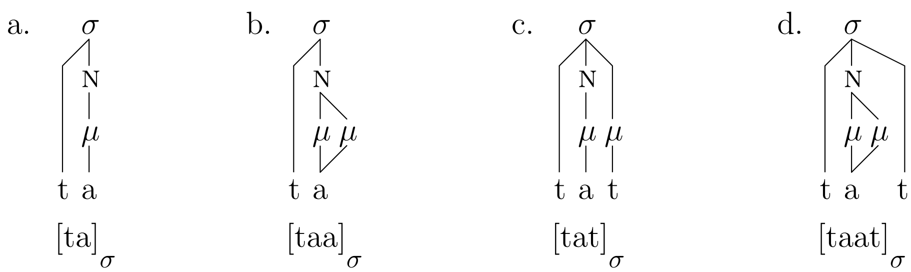
It appears, then, that positive circumscription of a base may appeal to syllables or feet. It has been observed, however, that total reduplication of a positively circumscribed syllable would amount to syllable copy, a presumably nonexistent phenomenon (Moravcsik 1978, Marantz 1982). Viewed in a slightly more detailed way, syllable copy would require a morphological operation (total reduplication, which is tautologous compounding McCarthy & Prince 1988:14) to appeal to a phonological entity (syllable) as if it were a morphological domain and then compound it. McCarthy & Prince (1988:6–7) note that total reduplication involves copy of entities such as Word, Root, Stem, or Phonological Word, all potentially morphological entities. In attested total reduplication with prosodic circumscription (such as the case of Yidin^{y} discussed in McCarthy & Prince 1991:232–234), the material circumscribed (Minimal Word) is a potential morphological entity (Word). For syllable copy, on the other hand, there would not be any corresponding morphological counterpart for the circumscribed material (syllable). This mismatch between copied material and morphological material may well be what rules out syllable copy. To continue to rule out the existence of syllable copy, then, we could posit the restriction in (135) in place of the stipulation of (134).

(135) Total reduplication may only copy material which is a potential morphological category.

This restriction will allow positive circumscription of a syllable while still ruling out syllable copy. In terms of (135), the lack of syllable reduplication is a fact about reduplication as an operation on morphological elements, not a fact about circumscription per se.

2.4.2.2 Nuclear Moraic Model

The Nuclear Moraic model of Shaw (1992, 1993) provides an interesting alternative account of reduplication in Southeastern Tepehuan. Under this view, a structural distinction is to be drawn between bi-moraic syllables consisting of a cvv sequence (136b) and those consisting of a cvc sequence (136c). The inclusion of the traditional Nucleus constituent provides the configurational distinction.

(136)


An open syllable is one that has the nucleus constituent at its right edge.

The template for Southeastern Tepehuan could be a bi-moraic open syllable (136b). As mentioned in section 2.1, underlying forms all have open syllables. It would be unsurprising that the reduplicative template would also be open. Per McCarthy & Prince (1990:231), a minimal word could then be safely circumscribed. A form like baaban from /banV/ (131a), would consist of circumscribing banV and mapping it to the bi-moraic open template. Only vowels can fill the nucleus constituent, so the template would be realized as baa.

Shaw (1992) discusses reduplicative evidence for all of the various templates predicted by the Nuclear Moraic model, except for the bi-moraic open template. Southeastern Tepehuan provides just such a case.

The Nuclear Model also provides another means to express the ClosedFtHd constraint. The syllable of the foot head should not end in a nucleus:

(137) ClosedFtHd: $* \;\text{N}]_{\sigma}\,]_{\text{Ft}}$

This can be compared with the earlier formulation, repeated in (138).

(138) ClosedFtHd: $* \text{V}]_{\text{Ft}}$ (=66)

The implication in both cases is that the foot should end in a syllable which is consonant final (i.e. closed). The theoretical challenge is that the concepts of "open" versus "closed" syllables are not true prosodic categories under either formulation. This remains a matter for future research.[39]

2.4.3 Exceptional Forms

While Stem Binarity does capture the generalization of the surface form, some lexical items need to be specially marked as to how that binarity is achieved. Consider the three forms in (139).

(139)

	base	singular	plural	glosses	
a.	/sui.ma.ri/	$[\text{sui}]_F[\text{mar̃}]_F$	$[\text{suis}]_F[\text{mar̃}]_F$	'deer (sg,pl)'	(=131h)
b.	/gio.tɨ.rV/	$[\text{gio}]_F[\text{tɨr}]_F$	$[\text{gio'n}]_F[\text{tɨr}]_F$	'plain(s)'	(=88c)
c.	/vai.nu.mV/	$[\text{vai}]_F[\text{ñum}]_F$	$[\text{va.pai}]_F[\text{ñum}]_F$	'metal(s)'	(=132e)

All three have the same prosodic shape in the base. The first two employ a bi-moraic template and then delete the entire nucleus of the second syllable to achieve binarity. The third apparently opts for a mono-moraic template and preserves the nucleus of the second syllable.

A few other forms appear to employ a mono-moraic template as shown in (140).

[39] See McCarthy & Prince (1993:118–119) for an interesting discussion along this line. The discussion crucially depends on the notion of optimization and parallel candidate sets.

(140)

		base	singular	plural	glosses	
a.	i.	/gaa.tV/	$[\text{gaat}]_F$	$[\underline{\text{ga}}.\text{gaat}]_F$	'bow(s)'	
	ii.	/koo.gV/	$[\text{koo'}]_F$	$[\underline{\text{koo}}]_F[\text{ko'}]_F$	'snake(s)'	(=131f)
b.	i.	/haa.ra.si/	$[\text{haa}]_F[\text{raš}]_F$	$[\underline{\text{ha}}.\text{haa}]_F[\text{raš}]_F$	'crab(s)'	(=132c)
	ii.	/pii.pi.ri/	$[\text{pii}]_F[\text{pir̃}]_F$	$[\underline{\text{pii}}\text{p}]_F[\text{pir̃}]_F$	'chick(s)'	(=131g)
c.	i.	/ka.rV.va.si/	$[\text{kar}]_F[\text{vaš}]_F$	$[\underline{\text{ka}}.\text{kar}]_F[\text{vaš}]_F$	'goat(s)'	(=132a)
	ii.	/na.ka.mɨ.ri/	$[\text{nak}]_F[\text{mɨr̃}]_F$	$[\underline{\text{naan}}]_F[\text{ka.mɨr̃}]_F$	'bat(s)'	(=131c)

The prosodic structure of the base is identical for each pair. In the (i) cases, a mono-moraic template is used, while in the (ii) cases the template is bi-moraic. I take it that the forms with a mono-moraic template need to be specially marked in the lexicon. Despite the lexical markedness of these cases, it is still the case that the output maintains stem binarity.

2.4.3.1 Exceptionality and Constraint-Ranked Derivation

Under the Constraint-Ranked Derivation approach, these exceptional forms simply employ a mono-moraic (i.e. core syllable σ_c) template rather than the default bi-moraic template in the underlying form. For example the derivation for (140a.i) would proceed as in (141).

(141)

Representation		Constraint	Repair	OK?
Underlying	/σ_c-gaa.tV/			
Reduplication	ga.gaa.tV			
Input	$[\text{ga.gaa}]_F[\text{tV}]_\sigma$	ParseC		
		StemBin		
		∗ClosedFtHd	↦ $[\text{ga.gaat}]_F$V	
	$[\text{ga.gaat}]_F$V	∗ParseV	N/A	
Output	$[\text{ga.gaat}]_F$V			
Surface	$[\text{ga.gaat}]_F$			

The template is lexically selected to be a core syllable and the reduplication process produces the structure represented in the "Reduplication" line. The stem well-formedness process has the input given in the "Input" line. This representation passes both ParseC and StemBin (it has configuration (90c)). The repair for the violation of ClosedFtHd closes the one and only foot, producing the attested form.

2.4.3.2 Exceptionality and Parallel Candidate Sets

While the treatment of such exceptionality is straightforward under the Constraint-Ranked Derivation approach, it is not immediately clear under the parallel candidate set approach. Both long and short versions of the reduplicated form must be included in the candidate set.[40] The constraints must be ranked to select the default case. How are such marked forms to be correctly selected?

The issue is not merely limited to reduplication. Recall that the forms in (142) do not meet StemBin.

(142)	a.	/ha-kuu.ku.sio.poo/	⟶	ha-[kuuk]$_F$[šio]$_F$[poo]$_F$	'their necks'	(=105a)
	b.	/koo.kV.roi.da.gV/	⟶	[kook]$_F$[roi]$_F$[ǰa']$_F$	'tadpole'	(=105b)
	c.	/koo.koo.kV.roi.da.gV/	⟶	[ka.kook]$_F$[roi]$_F$[ǰa']$_F$	'tadpoles'	(=105c)

These forms will also need to be (lexically) marked as exceptional to the StemBin constraint.

The function which creates the candidates is intended to be universal (McCarthy & Prince 1993:20). Therefore, no language specific information (such as exceptionality)

[40] According to McCarthy & Prince (1993:62), a candidate for the reduplicated material could consist of "any linguistic expression whatsoever." Thus, not only long and short versions must be considered, but also every other conceivable set of segments and prosodic structure, whether or not these segments bear any relevance to the base. Presumably, the term "linguistic expression" is to be limited to reasonable phonological expressions. Otherwise, a reduplicant candidate could consist of linguistic expressions such as s-structure trees or lambda expressions. These would clearly be inappropriate.

should be included in this function.

Universal Grammar also is posited to specify "... the set of constraints out of which grammars are constructed ... Individual grammars are constructed by imposing a ranking on the Universal constraint set, with some setting of parameters and fixing of arguments within the constraints" (McCarthy & Prince 1993:5). Since the constraints are intended to be general, lexical marking on a constraint does not seem appropriate, either.

The remaining component of the parallel candidate set approach is the "Best Satisfaction" (or Harmonic Ordering) algorithm. As was implied for the Constraint-Ranked Derivation approach for the StemBin exceptions in section 2.3.4.1 above, the algorithm itself could check for lexical exceptionality before evaluating the constraint. That is, the algorithm would check to see if the lexical entry declared constraint C_i inoperative.

The selection of template size, however, might not consist of "turning" off a constraint. Rather, it could be the selection of a different parameter for the template constraint. McCarthy & Prince (1993:139) give the following constraint schema:

(143) Constraint Schema for Classical Templates (McCarthy & Prince 1993:139)

Mcat=Pcat

where Mcat ≡ Morphological Category ≡ Prefix, Suffix, RED, Root, Stem, LexWd, etc.

and Pcat ≡ Prosodic Category ≡ Mora, Syllable (type), Foot (type), PrWd (type), etc.

The default template for the nominal plural in Southeastern Tepehuan would be as in (144).

(144) RED=$\sigma_{\mu\mu}$

For those forms exceptionally taking the core syllable template, the bi-moraic prosodic

category parameter could be substituted with a core syllable (σ_c).

Alternatively, both (144) and (145) below could be included in the constraint hierarchy, with (144) ranked (immediately) above (145).

(145) RED=σ_c

In the normal case, constraint (145) would always be violated. For a lexical exception case, constraint (144) would be "turned off." This would allow constraint (145) to become effective and select the appropriate candidate.

Either of these two conceptions would produce the desired result. The one which posits two templatic constraints might be preferred since it does not include any more machinery than would be needed for a case like the StemBin exceptions. The parameter-changing solution would require the algorithm to not only "turn off" a designated constraint, but also to modify the substance of a constraint.

2.5 Conclusion

The proposed analysis for the Southeastern Tepehuan nominal truncation patterns demonstrates how the Constraint-Ranked Derivation approach to optimization successfully and succinctly deals with the intriguing patterns. The traditional assumption about syncope being the deletion of the nucleus of the weak member of a metrical foot does not hold for Southeastern Tepehuan. Rather two configurational constraints provide the motivation for the truncations. A preference for closed foot heads governs many of the truncations, including the traditional notion of apocope. An overall constraint on the prosodic size of the stem explains the remaining patterns. This Stem Binarity constraint exemplifies another instance of the role of binarity within phonological systems.

Looking beyond Southeastern Tepehuan, let us consider the question whether the approach outlined in this chapter can cast some light on other cases of syncope in

the literature. For example, how does the essence of this approach carry over to the well-studied case of syncope in Tonkawa (see Kenstowicz & Kisseberth 1979:65–71, 431–433 and the references cited therein as well as McCarthy 1986:223–225)?[41] At first glance the forms in (146) appear to fall out from the Southeastern Tepehuan analysis.

(146)	a.	/picena-oʔ/	⟶	[pic]$_F$[noʔ]$_F$	'he cuts it'
	b.	/picena-n-oʔ/	⟶	[pic]$_F$[na.noʔ]$_F$	'he continually cuts it'
	c.	/we-picena-oʔ/	⟶	[wep]$_F$[ce.noʔ]$_F$	'he cuts them'
	d.	/picena/	⟶	[pi.cen]$_F$	'castrated one'

Every foot head is closed; every form is binary. A broader look, however, shows that there are significant differences. Not every foot head is closed when conceivably it could be (147a–c); the stem is not always binary (147a); and an initial long vowel will not remain long (147b).

(147)	a.	/we-picena-n-oʔ/	⟶	[wep]$_F$[ce.na]$_F$[noʔ]$_F$	'he continually cuts them'
	b.	/we-naat-oʔ/	⟶	[we.na]$_F$[toʔ]$_F$	'he steps on them'
	c.	/pile-n-oʔ/	⟶	[pi.le]$_F$[noʔ]$_F$	'he continually rolls it'
	d.	/we-pile-oʔ/	⟶	[wep]$_F$[loʔ]$_F$	'he rolls them'

As has been noted in the literature, Tonkawa displays a rule which deletes the second vowel of the stem as long as that vowel is not the final vowel of the stem (as in (147c)). We can view this as a process which removes the second mora of the stem and is subject to a stem binarity constraint as in (148).

(148) StemBin: Stems are minimally binary at the foot level.

This can be formalized as shown in (149).

[41] Only a preliminary sketch of an analysis will be given here. Issues such as antigemination and the OCP and what McCarthy (1986:224) calls Final Apocope are not dealt with.

(149) $$<\text{StemTruncation} : H> (R_{In}) \; = \; R_{Out},$$
$$H = \text{StemBin}$$

The StemBin constraint only functions to restrict the minimal size of the stem (rather than both the minimal and maximal size as in Southeastern Tepehuan). Furthermore, this binarity constraint only goes down to the foot level: minimally, the stem must be branching or, if there is only one foot, that foot must be branching. StemBin for Tonkawa would not have any associated repair strategy; it only serves to evaluate the representation against the minimal stem requirement.

To see how this would work, derivations are provided for (147c–d). The case where the deletion process passes StemBin (147d) is illustrated in (150).

(150)

	Representation	Constraint	Repair	OK?
Input	[we.pi]$_F$[loʔ]$_F$			
Process	[wep]$_F$i[loʔ]$_F$	StemBin		
Output	[wep]$_F$i[loʔ]$_F$			
Surface	weploʔ			

The stem is wepile. The final vowel is elided with suffixation to produce wepiloʔ. The mora-deletion process applies and removes the second mora of the stem. The resulting representation passes the StemBin constraint since the stem is prosodically binary (it is a part of two feet).

The minimal case of (147c) does not pass the minimality requirements of StemBin as shown in (151).

(151)

	Representation	Constraint	Repair	OK?
Input	[pi.le]$_F$[noʔ]$_F$			
Process	[pil]$_F$e[noʔ]$_F$	*StemBin		
Output	[pi.le]$_F$[noʔ]$_F$			
Surface	pilenoʔ			

The stem in this case is pile. The result of the mora deletion process produces a stem that does not meet StemBin; it is only one mono-syllabic syllable pil. This violation of StemBin causes the process to block. The output is therefore the original representation.

The proposed solution for Tonkawa provides another instance of the need for a process whose output is subject to well-formedness considerations. This is one of the key distinctions between the Constraint-Ranked Derivation approach and the parallel candidate set approach to optimization.

Under the parallel candidate set approach, the process must necessarily be accounted for via one or more constraints. The single constraint (or the effect of the multiple constraints) would insist on the non-parsing of the vocalic mora of the second syllable of the stem.[42] The constraint(s) would be ranked below StemBin.

Both the Southeastern Tepehuan and Tonkawa data illustrate the viability of the

[42] If stem truncation in Tonkawa is truly an arbitrary process, then this raises the question of the felicity of a phonology which disallows rules and insists on employing constraints to account for every aspect of the phonological system of a language. Constraints have a distinct advantage over rules since they provide an explanation for the observed patterns in the data. If the data truly contain inexplicable patterns, then requiring the use of constraints to account for such patterns weakens the explanatory advantage of constraints as analytical tools: some constraints would be explanatory and some would not. A theory which employs both rules and constraints, on the other hand, would be able to maintain the desired dichotomy: rules would account for the inexplicable patterns while constraints would provide explanations for the other patterns.

McCarthy (1993:183–187), working within the Optimality Theory framework, suggests that Boston English *r*-insertion should be handled via a rule. Such a rule would be an operation independent of the constraint system. He posits that the rule would create candidates with the appropriately inserted *r*.

While this has the advantage of maintaining the dichotomy between rules and constraints, we note that it removes the language independent nature of the generation function. One could conceive of the rules as being a separate module which applies to the output of the generation function, thereby maintaining the universality of the generation function. Such a module would always increase the size of the candidate set. For every candidate that meets the structural description of the rule, (at least) two candidates would be produced: the one that underwent the structural change and the one that did not. The grammar would then consist of universal and language specific constraints, the "Best Satisfaction" algorithm (which might include language specific information for exceptional forms — see section 2.4.3.2), the universal generation function, and the language specific set of phonological rules.

Constraint-Ranked Derivation approach to optimization. We have also shown how the parallel candidate set approach can account for the data. Both approaches maintain the key notion of optimization of ranked and violable constraints and its concomitant insights into the patterns of the data.

Chapter 3

Footing and Prominence Well-formedness in Pichis Asheninca Stress

3.1 Introduction

The previous chapter demonstrated the empirical adequacy of the Constraint-Ranked Derivation approach to optimization for issues of stem well-formedness in Southeastern Tepehuan nominal forms. This chapter focuses on a different aspect of prosody: stress assignment. In particular, it expands on recent work with respect to the nature of extrametricality in stress systems while exploring the relationship between foot headship and stress placement.

Many languages, including Classical Latin, have been analyzed as assigning stress by marking the final syllable as extrametrical (Hayes 1981, 1991:80). Since a language like Latin can contain mono-syllabic words which do bear stress, there is a conflict between extrametricality and stress assignment. Hayes (1991:92ff) refers to this as part of the "unstressable word syndrome" and Prince & Smolensky (1992) call it the "exhaustiveness property." Prince & Smolensky (1992) offer an explanation for this in terms of ranked and violable constraints.[1] There are two constraints in opposition to each other. One requires words to contain at least one foot and the other requires final syllables to be unfooted. Whenever the first constraint outranks the second so that the second is violated, "exhaustiveness" results. Mester (1994) has explicated

[1] Hayes (1991:47) accounts for this by including a "nonexhaustivity" clause in the theory of extrametricality; this clause blocks an extrametricality rule whenever that rule would render the entire domain extrametrical. Halle & Vergnaud (1987b:50) posit a conceptually similar condition.

how this idea can be applied to issues in Latin stress. This chapter will expand on Mester's work by further explicating the issues for an iambic stress system.

This stress system is the rather complicated one of Pichis Asheninca, a Maipuran Arawakan language spoken in the eastern high jungles of Peru. The initial work on stress in this language is Payne (1990), which lays out the intriguing data and motivates a thorough analysis based on the framework of Halle & Vergnaud (1987a). The only other work known to me on the stress of this language is Hayes (1991:246–253). He couches Payne's analysis in terms of his Parametric Metrical Theory.

As illustrated in (152), basic stress in Pichis Asheninca can be described as building iambic feet from left to right, with final syllables treated as extrametrical. (Stress is indicated by an acute accent (´) and syllables are separated by periods to aid in legibility.)[2]

---

[2]Payne (1990) gives the following consonantal chart:

(i)

	Bilabial	Alveolar	Palatal	Velar	Glottal
Stop	p	t		k	
Affricate		c			
Fricative		s			h
Nasal	m	n			
Flap		r			
Glide	w		y	ɣ	

Pichis Asheninca has four vowels:

(ii)

	Front	Back Unrounded	Back Rounded
High	i		
Mid	e		o
Low		a	

Permissible vowel clusters are the long forms aa, ee, ii, and oo and the diphthongs ai and oi.

The syllable structure of Pichis Asheninca requires an onset consonant except word initially. The only coda is the unspecified nasal N which must be place-linked with a following non-continuant. Therefore, no word ever ends in a syllable with a coda. A syllable may contain one or two vowels. The onset may consist of a consonant followed by the palatal glide. (There are some restrictions to this, but they do not concern us here.) Using a template as a descriptive device, the syllable in Pichis Asheninca is C(Y)V(V)(N), with the glide, second vowel and final N optional.

(152)	a.	no.kó.wa.wé.ta.ka	‘I wanted (it) in vain’
	b.	ha.má.naN.tá.ke.né.ro	‘he bought it for her’
	c.	pi.ñáa.páa.ke	‘you saw on arrival’
	d.	póo.ka.ná.ke.ro	‘you threw it out’

Di-syllabic words have stress on the penult, even if the final syllable is heavy:[3]

(153)	a.	há.ka	‘here’
	b.	syí.ma	‘fish’
	c.	hí.ñaa	‘water’

Given that final syllables are extrametrical, this is what one might expect. Footing considerations, however, raise several questions. In metrical grid conceptions of stress such as those of Halle & Vergnaud (1987b) and Hayes (1987, 1991), the relationship between foot head and stress is encoded directly on the grid by bracketing.[4] It is always the case that if a syllable bears stress, it is the head of its respective foot. If this is strictly true then the forms in (153) must either involve degenerate feet or be some kind of trochaic (i.e. left-headed) foot.

Positing degenerate feet is problematic for Pichis Asheninca. Hayes (1991:77–78) notes that no case has been found of an iambic system with degenerate words. Spring (1990a, 1990b) and Black (1991) demonstrate that the minimal word in the related dialect of Apurucayali Asheninca is at least bi-moraic. Given that the minimal word is precisely one foot (McCarthy & Prince 1990:231), it seems extremely unlikely that these forms consist of a degenerate foot plus a stray adjoined syllable.

If degenerate feet are to be avoided, then in order to keep a one-to-one correspondence between stressed syllable and foot head, we must allow a mixture of iambic and trochaic feet within this language. In order to account for (153c), it would be necessary to either posit syllabic trochees or posit a very strange moraic trochee ($\breve{\sigma}\bar{\sigma}$).

[3]Example (153c) is actually listed as jí.ñaa, a typographical error. The initial j is a carry over from the practical (Spanish-oriented) orthography. The ñ is equivalent to ny.

[4]See Liberman & Prince (1977) and Hammond (1984) for other means of expressing this relationship.

The first option would require mixing “quantity-sensitive” iambs with “quantity-insensitive” syllabic trochees. Not only would this involve a shift in foot headship from right-headed to left-headed, but it would also involve a shift in moraic weight distinctions since syllabic trochees ignore syllable weight while iambs do not.

The second option of positing a moraic trochee for (153c) is equally unsavory. One of the key characteristics of a moraic trochee is the maximally bi-moraic status of its feet (Hayes 1987, 1991:58). The trochee under consideration would be tri-moraic. In addition, it would violate the Weight-to-Stress Principle of Prince (1990) (if heavy, then stressed) because the heavy syllable would not bear stress.

Suppose, instead, that there actually is a canonical iambic foot in (153c). This solves all problems with respect to footing. There is neither a degenerate foot nor a change in foot type. Minimal word considerations are also clearly satisfied. The problem, of course, is that the syllable bearing stress is not the head of the foot. This is in direct conflict with standard conceptions of the metrical grid, where there is a one-to-one correspondence between foot headship and stress: if a syllable bears stress, it must be the head of its respective foot.

To deal with this situation, I suggest that the standard conception of the metrical grid be reconsidered and, further, that it be reconsidered in light of the Optimality Theoretic (Prince & Smolensky 1991, 1992, McCarthy 1993, McCarthy & Prince 1992, Prince & Smolensky 1993a, 1993b) notions of ranked and violable constraints as couched within the Constraint-Ranked Derivation approach to optimization. Recall that Constraint-Ranked Derivation views a process as operating on a representation to produce a tentative representation. This tentative representation is then subject to a hierarchy of ranked and violable well-formedness constraints. The resulting representation is the optimal representation given the hierarchy.

Suppose that the grid is an autonomous entity separate from, but related to, footing. This is somewhat similar in concept to the classical grid notion of Liberman

& Prince (1977) and Prince (1983), but marking on the proposed autonomous grid is still tied to foot structure.[5] As in the standard conception, a syllable bears stress if and only if it is so marked on this grid. There is a prominence marking process which always marks foot heads for prominence on this autonomous grid. In addition, there is a universal well-formedness constraint that insists that prominence be marked on foot heads. As long as this constraint is ranked relatively high in the hierarchy, the typical case of "if stressed, then foot head" will occur. A non-head will bear stress only under pressure from other well-formedness constraints. That is, the constraint requiring prominence on foot heads can be violated if it is outranked by some constraint that declares prominence on a particular foot head to be illformed.

I claim that such is the case for the forms in (153). Following the work of Prince & Smolensky (1992, 1993b:43ff) and Mester (1994) on extrametricality, I propose that one well-formedness constraint for Pichis Asheninca relates to the avoidance of final prominence on this grid. Thus, the forms in (153) all involve pure iambic footing with prominence being forced off the head onto the non-head to meet the requirement of non-final prominence.

Such prominence considerations are needed independently for Pichis Asheninca. Notice how the basic iambic pattern is perturbed in the forms in (154).

(154)	a.	i.	*ka.kí.ta.ke	
		ii.	ka.ki.tá.ke	'he/she woke up'
	b.	i.	i.pí.co.ka	'he turned around'
		ii.	i.pi.có.ka	
	c.		nò.syi.ya.pì.ca.tàN.ta.ná.ka.ri.ri	'I escaped from him'
	d.		iN.kìN.ki.syi.re.tà.ko.tà.wa.ké.ri	'he thought about it for a while'

The antepenult never bears stress in a form like (154a), yet it will optionally bear stress for a form like (154b). None of the final three light syllables bear stress in (154c)

[5] The proposal is thus also similar to the use of both metrical trees and metrical grids in Hayes (1984).

while the penult does in (154d). Furthermore, contrary to the normal iambic pattern, the initial light syllable in (154c) bears stress. If this evidences a trochaic foot while the rest of the word contains iambic feet, then there is a violation of the Uniformity Parameter (McCarthy & Prince 1986:10). This parameter requires languages to have the same foot type either for all words of the language or at least within a given word. A word is never allowed to mix foot types. In addition, there is a sequence of three unstressed light syllables (ki.syi.re) in (154d).

As insightfully noted by Payne (1990), these perturbations are based on sonority prominence issues, not on footing. The footing involves standard iambs. Following key insights of both Payne (1990) and Hayes (1991) and crucially employing the notions of Constraint-Ranked Derivation, I analyze the stress system of Pichis Asheninca as consisting of three steps:[6]

(155) i. The prosodic footing process determines the initial sites for prominence marking.
ii. The prominence marking process places grid marks on the autonomous grid for every foot head. The output of the prominence marking process is optimized according to a hierarchy of ranked and violable well-formedness constraints, many of which are sonority based.
iii. The main stress process combines footing and prominence in that it assigns main stress to the right-most prominence within the final two feet or colon of the word.

The optimization process may remove or shift prominence marks on the grid.

This chapter is organized as follows. Section 3.2 details the proposed modification to metrical theory and illustrates it from the familiar case of Latin. Section 3.3 motivates and illustrates the full account of stress assignment in Pichis Asheninca. This includes a demonstration that the optionality issues are a direct result of the application of a repair strategy to one of two possible repair sites. It also discusses

[6] With respect to (155-iii), see Hammond (1987) for other evidence of the role of the colon in prosody.

the implications this optionality has for the parallel candidate set approach to optimization. Issues of main stress placement are then given in section 3.4. Finally, section 3.5 discusses the distinctions between previous accounts of stress in Pichis Asheninca and the account posited here.

3.2 Footing and Prominence

Before dealing with the intricacies of Pichis Asheninca stress, I will explicate the proposed relation between footing and prominence on the grid within the Constraint-Ranked Derivation approach. As mentioned above, footing and stress assignment are viewed here as being split into two separate, but related processes. Such a separation is needed independently for prosodic morphology (McCarthy & Prince 1986, 1988, 1989, 1990, 1991) and stressless languages like Japanese that evidence foot structure (Poser 1990, Itô 1990, Mester 1990, Itô, Kitagawa, & Mester 1992, and Itô & Mester 1992).

The footing process is detailed in section 3.2.1. This is followed by an explanation of the prominence marking process in section 3.2.2.

3.2.1 The Footing Process

Footing involves the usual notions of building iambs, moraic trochees, or syllabic trochees in either a left-to-right or right-to-left direction (see Hayes 1987, 1991; Halle & Vergnaud 1987b posit a conceptually similar, yet different approach). For bounded systems, I assume the standard parametric conceptions: a particular language selects for foot type and direction. Unlike the standard conception, I posit that the representation so produced is then submitted to a hierarchy of well-formedness constraints.

Given word minimality (Prince 1980, McCarthy & Prince 1986:8), every prosodic word must contain at least one foot. In addition, prosodic licensing (Itô 1986, Goldsmith 1990, Itô & Mester 1993) requires each prosodic unit to belong to higher

prosodic structure. Prince & Smolensky (1992) have couched this in terms of the constraint given in (156).[7]

(156) LW=PW: A lexical word must contain a prosodic word.

This requires the footing process to foot every lexical word. Other constraints may be needed on a language particular basis.

As mentioned in section 3.1, languages which have final syllable extrametricality can have the exhaustiveness property. To deal with this situation, Prince & Smolensky (1992) employ a constraint such as the one in (157).[8]

(157) FinFt: $\ast\ \begin{array}{c} \text{F} \\ | \\ \sigma]_{\text{Wd}} \end{array}$

This avoids footing a word final syllable. A key insight of Prince & Smolensky (1992) is that the LW=PW constraint requires the representation to contain a foot. As a high-ranked constraint, it provides a higher motive over the stress-oriented constraint of FinFt, and thus "revokes" extrametricality. They maintain that revocation of extrametricality is not a general property of extrametricality per se. They cite examples from tone (Hewitt & Prince 1989) and the prosodic morphology view of infixation (McCarthy & Prince 1986:44ff, 1990:227ff) where there is not any revocation of extrametricality.

This constraint is a very reasonable one in a non-derivational model, where the best candidate representation is selected from among a set of representations. FinFt rejects any candidate representation that has a final footed syllable.

The constraint is not so felicitous, though, in a derivational model like the one

[7] Prince & Smolensky (1992) assume that a prosodic word contains a foot. See also Prince & Smolensky (1993b:43).

[8] They call the constraint NonFinality. I use FinFt as an abbreviation for "Final Foot." Mester (1994) also employs this constraint for Latin, ranking it below one that avoids a foot head in final position. See also Prince & Smolensky (1993b:43) for further development of this idea.

assumed here. Consider the case where feet are built from right-to-left. The footing process would necessarily foot the final syllable of the input representation in every case. The constraint would usually be violated and would force the input to be repaired. The correct repair would involve skipping the final syllable. Such a scenario is not attractive because the footing process would normally produce an illformed representation.

There is a solution, however. The prosodic circumscription view of an operation applying under extrametricality (McCarthy & Prince 1990:226) involves the application of the operation to a base which has had a prosodic constituent circumscribed from an edge. Final syllable extrametricality is a case where a syllable is circumscribed off the right edge of the base and the footing operation is applied to the remaining material.

Suppose we take this view of footing and place it in the Constraint-Ranked Derivation model. The footing process applies to a representation consisting of a sequence of syllables. The final syllable is circumscribed and feet are built from right-to-left on the rest of the representation. The final syllable is then concatenated back onto the modified representation.[9] The resulting output representation will thus observe final syllable extrametricality. Per the Constraint-Ranked Derivation view, this representation is then passed through a hierarchy of ranked and violable constraints. Suppose that there is but one such constraint, namely LW=PW. Further, suppose that the repair strategy associated with this constraint is "impose a foot" (Mester 1994). This is a very natural repair strategy for a constraint that insists a word should have a foot.

This view achieves two positive results. First, it empirically gets the facts right.

[9]Prosodic Circumscription Theory is re-examined in the parallel candidate set approach to optimization since this approach does not allow operations. See McCarthy & Prince (1993:102ff) for discussion.

Secondly, it theoretically maintains the insights of Prince & Smolensky (1992). Revocation of extrametricality is not associated with the extrametricality operation (i.e. circumscription). Rather, it is associated with footing well-formedness.

I will illustrate this using the Latin forms in (158).[10] Latin builds moraic trochees from right-to-left with the final syllable extrametrical (Hayes 1991:80, who attributes the essential idea to Allen 1973; also see Mester 1994). In the view taken here, no degenerate feet are allowed; every moraic trochee must be bi-moraic.[11]

(158) a. [á.ni]<ma> 'soul'
b. [rḗ] 'thing'
c. [lé.ge] 'read, imp.'
d. [bé.ne] 'good' (from /benē/)
e. *re (non-existent form)

The derivation for (158a) is given in (159). It shows the schematic representations for the syllabification, circumscription, and footing processes. A foot is indicated by square brackets ([ ]). The output of footing is checked against the LW=PW constraint and repaired as needed. The final syllable is circumscribed, so the footing operation only applies to the sequence a.ni.

(159)

	Representation	Constraint	Repair	OK?
Syllabification	a.ni.ma			
Circumscription	a.ni			
Footing	[a.ni]ma			
		LW=PW		
Result	á.ni.ma			

The representation resulting from footing has the final syllable concatenated on to the end. Since there is a foot, LW=PW is satisfied.

The next four derivations involve violations of LW=PW. A mono-syllabic heavy

[10] The forms are taken from Mester (1994). A vowel with a macron (¯) indicates a long vowel.

[11] In the notation of Mester (1994), the strictly bi-moraic quantitative trochee $\mathrm{QT}<2,2>$ is built.

(158b) is derived in (160).

(160)

	Representation	Constraint	Repair	OK?
Syllabification	rē			
Circumscription				
Footing	rē			
		∗LW=PW	↦ [rē]	√
Result	rḗ			

Circumscription removes the one and only syllable from the purview of the footing operation. Hence, no foot is built as shown in the "Footing" line. This, of course, violates LW=PW. As its repair strategy, a foot is imposed on the sequence rē as shown to the right of LW=PW. Since this now meets the requirements of LW=PW, it becomes the representation of choice. This produces the correct result.

A similar violation occurs for (158c) as shown in (161).

(161)

	Representation	Constraint	Repair	OK?
Syllabification	le.ge			
Circumscription	le			
Footing	le.ge			
		∗LW=PW	↦ [le.ge]	√
Result	lé.ge			

This time circumscription leaves a lone light syllable. No legitimate bi-moraic foot can be built as shown, which results in a violation of LW=PW. Again, the repair strategy imposes a foot on the entire sequence le.ge and produces a well-formed foot. The constraint is thus satisfied.

The next case is one of "Iambic Shortening," where a final long vowel is shortened.

(162)

	Representation	Constraint	Repair	OK?
Syllabification	be.nē			
Circumscription	be			
Footing	be.nē			
		∗LW=PW	↦ [be.ne]	√
Result	bé.ne			

As in (161), footing is not able to produce a foot on the base material after circumscription. The violation of LW=PW involves the imposition of a foot on the entire sequence be.nē. As Mester (1994:14–15) argues, the imposition of a bi-moraic foot in this case results in shortening the final long vowel; the designated repair strategy of REMOVE-μ applies to the final vowel.[12]

The final case (158e) is for an unattested form which consists of a single light syllable.

(163)

	Representation	Constraint	Repair	OK?
Syllabification	re			
Circumscription				
Footing	re			
		∗LW=PW	↦ re	∗
Result	(none)			

No foot is built so LW=PW is violated. The repair of imposing a bi-moraic foot cannot be made, resulting in the failure of the representation to pass LW=PW. The

[12]Mester (1994:15–16) argues that the reason we do not get be[nē] is that extrametricality involves not only an avoid footing constraint, but also an avoid foot-head constraint for final syllables. The latter is ranked higher and rules out the be[nē] case. (All this relates to the notion that ends of words are prosodically weak positions.) Since in the view taken here, extrametricality is just a function of prosodic circumscription during the footing operation, this is accounted for by insisting that the repair strategy for LW=PW imposes a foot on the entire form.

If I understand the facts correctly, Iambic Shortening was optional to some extent. There could be two ways to repair such a violation of LW=PW: (i) impose a foot on the entire sequence with concomitant shortening of the final long or (ii) impose a foot with a resulting $\breve{\sigma}\bar{\sigma}$ trochee.

Constraint-Ranked Derivation algorithm (22) in chapter 1 declares that the process is blocked in such cases. The result is that a mono-moraic syllable is not even footed. A constraint such as Prosodic Licensing (Itô 1986, Goldsmith 1990, Itô & Mester 1993) would prevent the form from occurring. The proposed analysis, then, correctly predicts that such a form will not occur.

3.2.2 The Prominence Marking Process

Having discussed the footing process, we now turn to the prominence marking process. After feet are built and optimized, prominence is projected onto the autonomous metrical grid. For each foot head, the prominence marking process places a mark on the grid. The resulting representation is then submitted to a hierarchy of prominence oriented well-formedness constraints.

There are at least four such constraints. The first two are universal and do not have any associated repair strategies:

(164) a. LwProm: A lexical word has prominence on the grid.
b. HdProm: Foot heads have prominence on the grid.

LwProm is the prominence side of the LW=PW constraint of Prince & Smolensky (1992). It reflects the fact that every lexical word must have some prominence on the grid in order to bear stress.[13] HdProm asserts that foot heads should have prominence on the grid.

The other two constraints are language specific. For languages which observe a quantity distinction, Prince (1990:358) posits the Weight-to-Stress Principle (WSP), which I modify as in (165).[14]

[13] Languages such as Japanese that do not have stress will not have a prominence marking process. The LwProm constraint will therefore not apply in these languages.

[14] Prince originally used the word "stressed" instead of "prominence on the grid." The term was thus purposely ambiguous between "foot head" and "prominent on the grid." Under the conception of stress placement posited here, "stressed" can only mean "prominent on the grid."

(165) WSP: If a syllable is heavy, then it has prominence on the grid.

The WSP constraint asserts that if a syllable is heavy (i.e. bi-moraic), then it should bear stress (i.e. have prominence on the metrical grid). Like LwProm and HdProm, WSP does not have an associated repair strategy.

For "quantity sensitive" binary footing, WSP and HdProm do overlap. Heavy syllables are necessarily foot heads. This does not entail, however, that these two constraints are the same. We will see a case where they need to be separately ranked in section 3.3.1.2.

Some languages may also employ the constraint in (166).[15]

(166) FinProm: $* \overset{+}{\sigma}]_{\mathrm{Wd}}$

That is, avoid prominence on the final syllable. This reflects the insight of Mester (1994:16) that final syllable extrametricality involves the avoidance of placing the head of a foot in word final position.

Both Mester (1994) and Prince & Smolensky (1992, 1993b:43ff) deal with final syllable extrametricality in terms of foot parsing. This works quite well for a trochaic system where foot heads are final only for heavy syllables. For a moraic trochee case like Latin where final syllables are extrametrical, final syllables are footed only under minimality considerations as we saw in section 3.2.1. Furthermore, a di-syllabic light-light form ($\breve{\sigma}\breve{\sigma}$) never has its head word finally. For iambic systems, however, foot heads will be word final for both di-syllabic ($\breve{\sigma}\sigma$) and mono-syllabic ($\bar{\sigma}$) feet. As we saw in section 3.1, final syllable extrametricality for an iambic system such as Pichis Asheninca leads to difficulties with foot structure. By ascribing the effect

[15] Partly for aesthetic reasons, I use the symbol + to indicate prominence rather than the asterisk (∗) of Halle & Vergnaud (1987a, 1987b) or the x of Prince (1983) and Hayes (1987, 1991). In addition, an asterisk is also conventionally used to label a form as "unattested" and is used as an indication of a constraint violation. The use of a different symbol avoids any potential confusion in the derivation tables employed here.

of final light syllable extrametricality to prominence, we avoid such difficulties while still maintaining the results for trochaic systems.

Unlike the other three constraints, FinProm does have an associated repair strategy. To avoid redundancy between the constraint and the repair strategy, I follow the pattern established by Mester (1994) for prosodic repair strategies,[16] and posit two prominence repair strategies.

(167) Prominence Repair Strategies

a.	SHIFT-+	shift prominence within foot
b.	REMOVE-+	remove prominence from foot

The first repair strategy, SHIFT-+, is invoked for constraints which focus on prominence within a single foot. Since FinProm is such a constraint, its designated repair strategy is SHIFT-+. The shifting operation is defined so as to cause a prominence mark on a foot head to shift to the non-head and a prominence mark on a non-head to shift "off the foot".[17] The latter is the same as removal of the prominence mark.[18] The second repair strategy, REMOVE-+, is invoked for constraints which focus on prominence marks across foot boundaries, as in "stress clash." We will see several cases of the need for REMOVE-+ in section 3.3.1.

The Pichis Asheninca cases of di-syllabic words, repeated from earlier in (168), will be used to illustrate how the proposal works.

[16]See (26) and surrounding discussion in chapter 1 as well as the discussion about repairs for the ClosedFtHd and StemBin constraints in chapter 2.

[17]SHIFT-+ is similar to the "Move-x" operation of Prince (1983:33) and the "Beat Movement" grid euphony rule of Selkirk (1984b:55), except that SHIFT-+ is subject to foot structure. REMOVE-+ is similarly akin to the "Beat Deletion" grid euphony rule of Selkirk (1984b:56).

[18]Alternatively, one could just apply SHIFT-+ first; if this failed to improve the representation, then one could apply REMOVE-+. This parallels Mester's Prosodic Repair Strategies. For the Latin cases studied by Mester, however, every constraint could be associated with the designated repair strategy. The secondary repair mechanism is invoked only when the primary one fails to improve well-formedness. For the clash cases in sections 3.3.1.2 and 3.3.1.3, such an approach will fail to achieve the correct forms.

(168) a. há.ka ‘here’ (=153a)
b. hí.ñaa ‘water’ (=153c)

The four constraints need to be ranked with respect to each other. The universal LwProm is ranked highest since it is inviolable. In order to prevent prominence from ever occurring on a final syllable, FinProm must be ranked above both WSP and HdProm. I will arbitrarily rank WSP above HdProm for now as shown in (169).

(169) $<$Prominence Projection : $H>$ $(R_{In}) = R_{Out}$,
where H = LwProm ≫ FinProm ≫ WSP ≫ HdProm

The footing process produces iambic feet. Prominence will be marked on the foot heads and the resulting representation will be passed through the ranked constraints. A sample derivation for (168a) is given in (170).

(170)

	Representation	Constraint	Repair	OK?
Input	· + [ha.ka]	LwProm		
		*FinProm	↦ + · [ha.ka]	√
	+ · [ha.ka]	WSP		
		*HdProm	N/A	
Output	+ · [ha.ka]			

The autonomous metrical grid is shown above each syllable. Each syllable has a position on the grid. Syllables lacking prominence are marked by a period as a place-holder.[19] Prominence is indicated by the prominence symbol (+). The “Input” line shows the footing with prominence marked on the head. FinProm is violated because the final syllable is marked for prominence. Its designated repair strategy is shift-+ which produces the representation shown to the right of FinProm. Since this representation passes all higher and equal constraints, it becomes the represen-

[19]This is patterned after the bracketed grid notation of Hayes (1987, 1991).

tation of choice. Even though the representation violates HdProm, it is the optimal representation given the hierarchy.

The derivation for (168b) is similar.

(171)

	Representation	Constraint	Repair	OK?
Input	[$\overset{\cdot}{\text{hi}}.\overset{+}{\text{ñaa}}$]	LwProm		
		∗FinProm	↦ [$\overset{+}{\text{hi}}.\overset{\cdot}{\text{ñaa}}$]	√
	[$\overset{+}{\text{hi}}.\overset{\cdot}{\text{ñaa}}$]	∗WSP	N/A	
		∗HdProm	N/A	
Output	[$\overset{+}{\text{hi}}.\overset{\cdot}{\text{ñaa}}$]			

The only difference is that both WSP and HdProm are allowed to be violated.

By separating footing and prominence marking in this way, the minimal word cases of Pichis Asheninca are accounted for without having to posit "odd" foot structures. This analysis eschews degenerate feet as well as either mixing syllabic trochees with iambs or positing a highly marked moraic trochee.

3.3 Well-Formed Prominence

We now address the full range of prominence oriented stress perturbations in Pichis Asheninca. As we will see, there are factors in the Pichis dialect which led Payne (1990) to posit four distinct syllable weights as delineated in (172).

(172)

Class	Syllable patterns
Extra-light	$\underline{\text{syi}}$ and $\underline{\text{ci}}$ only
Light	other Ci
Normal	CiN and CV(N), where V is either $\underline{\text{a}}$, $\underline{\text{e}}$, or $\underline{\text{o}}$
Heavy	CVV(N)

The standard moraic split, of course, is between heavy and all the rest. The distinction between the other three is primarily a function of relative sonority (Payne 1990:194). The extra-lights are a special category peculiar to Asheninca.

Hayes (1991:231) observes that what has traditionally been referred to as "syllable weight" is actually a consequence of two factors. The first, syllable quantity, reflects the time dimension of syllable weight, whereby canonical iambs have uneven duration and canonical trochees have even duration.[20] The second involves syllable prominence or perceptual saliency. For some languages, the relative loudness of a particular syllable may play a role in the phonological stress system.

As Hayes (1991:246) notes, such is the case for Pichis Asheninca. Syllable quantity (moraic structure) delimits the footing in a straightforward fashion. The complexities arise with respect to how prominence is intertwined with the basic foot structure.

In this section, several prominence oriented well-formedness constraints are motivated in section 3.3.1. Issues relating to optionality and the parallel candidate set approach to optimization are discussed in section 3.3.2.

3.3.1 Prominence Constraints

Four constraints specific to Pichis Asheninca are presented in a stepwise fashion. Two resolve clash; one reflects sonority; and the other resolves clash via sonority.

3.3.1.1 Clash between Head and Non-Head

The forms in (173) all lack stress on the final two syllables.

(173)	a.	ka.máN.ta.ke	'he/she said'	
	b.	no.kó.wa.wé.ta.ka	'I wanted (it) in vain'	(=152a)
	c.	póo.ka.ná.ke.ro	'you threw it out'	(=152d)

[20] This idea is explored and illustrated in Hayes (1985, 1987) as well as Hayes (1991:70ff). An alternative view is presented in Kager (1993).

Under the assumptions pursued here, the two last syllables in all these forms are actually footed; only the prominence projected for the foot has been removed. To see why this should be so, consider the partial derivation for (173a) provided in (174).

(174)

	Representation	Constraint	Repair	OK?
Input	$[\overset{\cdot}{\text{ka}}.\overset{+}{\text{maN}}][\overset{\cdot}{\text{ta}}.\overset{+}{\text{ke}}]$	LwProm		
		*FinProm	$\mapsto [\overset{\cdot}{\text{ka}}.\overset{+}{\text{maN}}][\overset{+}{\text{ta}}.\overset{\cdot}{\text{ke}}]$	$\surd$

The initial prominence projection places prominence on the foot heads. FinProm is violated and its designated repair strategy, shift-+, moves the prominence marking to the non-head position. This places two prominence markings side-by-side, resulting in a clash between the prominence markings of the two feet.

The resolution of clash here is what one would expect: the non-head prominence is removed. A general prominence well-formedness constraint which insists that prominence should be on the foot head in a clash situation is given in (175). (This particular formulation assumes iambic feet.)[21]

(175) NonHdCl: $* \; \overset{+}{\sigma}][\overset{+}{\sigma}\ \overset{\cdot}{\sigma}]$

That is, avoid non-head prominence clash. This means that a representation is ill-formed when a foot head has prominence and the following syllable in non-head position also has prominence. This reflects the notion that prominence should be projected on foot heads. Note that while two consecutive heavy syllables will have adjacent prominence markings, they are not pronounced illformed by this constraint since neither has prominence in a non-head position; both are the heads of their respective feet. The repair strategy for NonHdCl is the designated one for clash

[21] Ideally, this would fall out from a constraint such as HdProm and some kind of clash avoidance constraint. Unfortunately, such an analysis does not account for all of the data. Non-heads do survive clash in certain cases as will be demonstrated in section 3.3.1.3. On the positive side, NonHdCl preserves the notion that clash is resolved by remove-+ and that it is always the non-head that is removed.

resolution: REMOVE-+.

The NonHdCl constraint must be ranked below FinProm. I also arbitrarily place it below WSP as shown for (173a) in the fuller derivation of (176).

(176)

	Representation	Constraint	Repair	OK?
Input	[ka.maN][ta.ke]	LwProm		
		*FinProm	↦ [ka.maN][ta.ke]	√
	[ka.maN][ta.ke]	WSP		
		*NonHdCl	↦ [ka.maN][ta.ke]	√
	[ka.maN][ta.ke]	*HdProm	N/A	
Output	[ka.maN][ta.ke]			

The resolution of the clash produces a representation that is optimal with respect to the hierarchy.

3.3.1.2 Clash between Two Heads

Another clash results when a foot consisting of two light syllables ($\breve{\sigma}\breve{\sigma}$) is immediately followed by a heavy syllable. The crucial sequence is underlined in (177).

(177) a. i. i.máN.ci.yá.wái.ti ‘he is sick’
ii. i.máN.ci.ya.wái.ti
b. i. a.tí.ri.pá.yée.ni ‘people’
ii. a.tí.ri.pa.yée.ni

The head of the di-syllabic foot may optionally bear stress. The key distinction between the optional clash resolution here and the obligatory clash resolution with NonHdCl is that here both syllables are heads. There is not as much pressure to resolve the clash as there is between a head and a non-head.

The constraint in (178) identifies such clashes between heads.[22]

(178) HdHdCl: $* \overset{+}{\sigma}][\overset{+}{\sigma}]$

That is, avoid clashes between heads. Since it is a clash oriented constraint, its designated repair strategy is remove-+. The constraint pinpoints where well-formedness is violated. The site of the repair could be either the left head or the right head; removal of prominence for either would resolve the clash. When the left repair site is selected, remove-+ removes the prominence mark and all higher and equal constraints will be satisfied. When the right repair site is selected, however, remove-+ will remove prominence from a heavy syllable. If the WSP constraint is ranked higher than HdHdCl, the repair will block because it will violate the higher constraint WSP. I suggest that this ambiguity in determining the repair site is the source of the optionality.

The derivation for (177b) is presented in (179). While WSP must be ranked above HdHdCl, I have also placed WSP above NonHdCl. This keeps the clash oriented constraints together. After the third clash constraint is introduced in section 3.3.1.3, the three constraints will be folded into one constraint. HdProm must remain low in the ranking; otherwise it would block the removal of prominence from a head.

[22]Another possible solution would be to optionally foot a $\breve{\sigma}\breve{\sigma}\bar{\sigma}$ sequence as $\breve{\sigma}[\breve{\sigma}\bar{\sigma}]$. The initial light syllable would be skipped. In the related language Apurucayali Asheninca (Payne, Payne & Sanchez Santos 1982), such sequences are always stressed as $\breve{\sigma}\breve{\sigma}\acute{\bar{\sigma}}$ unless the second light is an extra-light syllable (see section 3.3.1.4). In these cases, the sequence is stressed as $\acute{\breve{\sigma}}\breve{\sigma}\acute{\bar{\sigma}}$ suggesting that the footing was the usual $[\breve{\sigma}\breve{\sigma}][\bar{\sigma}]$. Unfortunately, there are no cases of a light, extra-light, heavy sequence in the Pichis Asheninca data. As mentioned in section 3.1, attempting such foot-oriented solutions to the Pichis Asheninca data poses serious problems. For these reasons, a prominence oriented analysis is pursued here.

(179)

	Representation	Constraint	Repair	OK?
Input	· + · + + [a.ti][ri.pa][yee]ni	LwProm FinProm WSP NonHdCl		
Option i.		∗HdHdCl	↦ · + · · + [a.ti][ri.pa][yee]ni	√
	· + · · + [a.ti][ri.pa][yee]ni	∗HdProm	N/A	
Output i.	· + · · + [a.ti][ri.pa][yee]ni			
Option ii.		∗HdHdCl	↦ · + · + · [a.ti][ri.pa][yee]ni	∗
	· + · + + [a.ti][ri.pa][yee]ni	HdProm		
Output ii.	· + · + + [a.ti][ri.pa][yee]ni			

Footing and prominence are assigned as shown in the "Input" line. There is a clash between pa and yee. Either head can be identified as the potential repair site. The first option shows pa as the repair site. remove-+ successfully removes the clash without violating any higher or equal constraints. The second option selects yee as the repair site. When prominence is removed from this head, however, it results in a violation of WSP; hence, the repair is blocked.

This analysis predicts that when the prominence on two consecutive heavy syllables clash, neither will be repaired. To remove prominence on either one would entail a violation of WSP since WSP is ranked higher than HdHdCl. The forms in (180) demonstrate that this prediction is correct.

(180) a. pi.ñáa.páa.ke 'you saw on arrival'
b. i.kyáa.píiN.ti 'he always enters'

3.3.1.3 Heads with <u>i</u> in Clash

There is yet another kind of clash in Pichis Asheninca as exhibited in (181).

(181) a. no.<u>pí</u>.to 'my canoe'
b. syoN.<u>kí</u>.ri 'type of partridge'
c. o.<u>pi</u>.ná.ta 'it costs'
d. ka.<u>ki</u>.tá.ke 'he/she woke up' (=154a-ii)
e. ka.<u>ri</u>.ná.ri 'colored'

Stress placement in (181a–b) is exactly where one would expect: on the head of the only foot. In (181c–e), however, the initial foot does not bear any stress. Stress surfaces on the following syllable, which is the non-head of the final foot.

Section 3.3.1.1 illustrated similar cases where the resolution to FINPROM produced a clash between the head of the initial foot and the non-head of the final foot. Some representative examples are repeated in (182).

(182) a. ka.máN.ta.ke 'he/she said' (=173a)
b. no.kó.wa.wé.ta.ka 'I wanted (it) in vain' (=152a)
c. póo.ka.ná.ke.ro 'you threw it out' (=152d)

The forms in (181c–e) have a similar clash situation as demonstrated by the partial derivation in (183) for (181d).

(183)

	Representation	Constraint	Repair	OK?
Input	[ka.ki][ta.ke]	LWPROM		
		*FINPROM	↦ [ka.ki][ta.ke]	√

What makes forms like (181c–e) unusual is that the head loses prominence rather than the non-head.

There is more to be considered as the forms in (184) exhibit.

(184) a. i. i.pí.co.ka ‘he turned around’ (=154b)
ii. i.pi.có.ka
b. i. ka.wí.ni.ri ‘cinnamon’
ii. ka.wi.ní.ri
c. i. i.kí.te.ti ‘he is clean’
ii. i.ki.té.ti
d. i. o.kí.co.ki ‘seed’
ii. o.ki.có.ki

These also involve a similar clash as evidenced by the partial derivation of (184c) shown in (185).

(185)

	Representation	Constraint	Repair	OK?
Input	$[\overset{\cdot}{\text{i}}.\overset{+}{\text{ki}}][\overset{\cdot}{\text{te}}.\overset{+}{\text{ti}}]$	LwProm		
		*FinProm	$\mapsto [\overset{\cdot}{\text{i}}.\overset{+}{\text{ki}}][\overset{+}{\text{te}}.\overset{\cdot}{\text{ti}}]$	$\surd$

In these cases, one prominence, either the left or the right, is removed. Which is removed appears to be optional.

Payne (1990) insightfully accounts for this kind of distribution by noting the relative sonority of the vowels involved. The $\underline{\text{i}}$ vowel is clearly deficient in some sense: it always gives way to a following $\underline{\text{a}}$ (181c–e), but is only optionally removed with a following non-low vowel (184). When non-high vowels are in the left position of such a clash, the right prominence is always removed as was shown in (182).

I will couch these insights in terms of the model posited here as follows. First, since the prominence of the high vowel is not as stable as the others,[23] the prominence well-formedness constraint in (186) is added to the hierarchy.

(186) iClash: $*\ \overset{+}{\text{i}}][\overset{+}{\sigma}$

[23]For main stress placement, $\underline{\text{i}}$ stands alone against $\underline{\text{a}}$, $\underline{\text{e}}$, and $\underline{\text{o}}$. That is, the prominence status of the mid vowels is more like the low vowel for main stress assignment, but more like the high vowel for prominence well-formedness. See section 3.4 for more discussion.

That is, having prominence on an i followed immediately by prominence on another syllable is to be avoided. iClash assumes that the i is the sole member of the nucleus; that is, syllables such as kii and kiN do not violate iClash. The following syllable may be either the head or the non-head of its foot. Since iClash is a clash oriented constraint, its designated repair strategy is remove-+.

Sonority plays a key role in determining the repair site. If the syllable following the i head has greater sonority, then the less sonorous i head is the locus for repair. If the following syllable is equal in sonority, then either site may be selected. The first case accounts for the obligatory removal of prominence on an i when it is followed by a light syllable with an a nucleus.

The latter case results in the optionality effects. Prominence will be removed on one of the two syllables involved in the clash. This conception of optionality is to be distinguished from the usual notion of an optional rule. For an optional rule, the structural change may or may not apply whenever the structural description is met. For iClash, the structural change always applies; the optionality is a result of the ambiguity of the location of the repair site.[24]

This proposal, of course, requires the sonority ranking to be specified appropriately. The proposed ranking is given in (187).

(187) Prominence Well-formedness Sonority Scale

$$\underset{\text{i,e,o}}{\breve{\sigma}} < \underset{\text{a}}{\breve{\sigma}} < \bar{\sigma}$$

For the prominence oriented well-formedness considerations posited here, it is crucial that e and o be ranked equally with i in sonority. As we will see in section 3.4, main stress assignment ranks the mid vowels with the low vowel instead. This difference in sonority can be nicely captured in traditional feature terms:

[24] This source of optionality is quite similar to the treatment of optionality in the Tokyo Japanese ŋ-rule of Itô & Mester (1989).

(188) iCLASH

`-low`	i,e,o
`+low`	a

Main Stress

`+high`	i
`-high`	a,e,o

iCLASH focuses on the feature `low` while main stress focuses on `high`. This is essentially the same solution as proposed by Payne (1990) and Hayes (1991), though couched in different terms. See section 3.5 for more discussion.

This ranking predicts that if a crucial i syllable is followed by a heavy syllable, then the i syllable will lose its prominence. The forms in (189) demonstrate that this is correct. The crucial feet involved are underlined.

(189) a. oN.ki.tái.ta.má.na.ke 'in the morning'
b. kaN.ti.mái.ta.cya 'however'

The iCLASH constraint must be ranked with respect to the other constraints. It must be ranked lower than FINPROM in order for the clash to arise in cases like those in (181c–e) and (184). For situations where the i syllable is followed by a non-head, iCLASH conflicts with NONHDCL. When the following syllable is a head, iCLASH conflicts with HDHDCL. To resolve such difficulties, Prince & Smolensky (1993a) posit the following theorem:[25]

(190) Pāṇini's Theorem on Constraint-ranking. (Prince & Smolensky 1993a)
Let C_1 and C_2 be constraints which conflict and which are both seen to be operative in some grammar. Then, if C_1 is more specific than C_2, it must be the case that in the grammar $C_1 \gg C_2$.

Clearly, iCLASH is more specific than the union of the constraints NONHDCL and HDHDCL. Therefore, iCLASH must be ranked above both of them. The ranking becomes that of (191).

[25]Prince & Smolensky (1993a) actually define the terms *operative* and *specific* in terms of candidate sets. Such sets are non-existent in Constraint-Ranked Derivation. The concepts behind the terms will have their usual meanings in a derivational model. Prince & Smolensky (1993b:81–82) provides the same Theorem, but in less perspicuous form.

(191) LwProm ≫ FinProm ≫ WSP ≫
iClash ≫ NonHdCl ≫ HdHdCl ≫ HdProm

The ranking can be simplified by folding the three clash constraints into one. This follows work by Prince & Smolensky (1992, 1993b:40ff) on Kelkar's Hindi in which two ranked constraints ("non-final" and "rightmost") are subsumed by a single constraint, "position". For the Pichis Asheninca stress system, the constraint Clash will subsume the three clash constraints as shown in (192).

(192) Clash: iClash ≫ NonHdCl ≫ HdHdCl

The resulting overall constraint ranking is given in (193).

(193) LwProm ≫ FinProm ≫ WSP ≫ Clash ≫ HdProm

Sample derivations are provided for the forms in (194).

(194)	a.	i.	ka.ki.tá.ke	'he/she woke up'	(=154a)
		ii.	*ka.kí.ta.ke		
	b.	i.	i.kí.te.ti	'he is clean'	(=184c)
		ii.	i.ki.té.ti		
	c.	i.	ka.máN.ta.ke	'he/she said'	(=173a)
		ii.	*ka.maN.tá.ke		
	d.	i.	kaN.ti.mái.ta.cya	'however'	(=189b)
		ii.	*kaN.tí.mái.ta.cya		

All three clash constraints will be listed in the derivations for perspicuity. The first is for a case of i-clash where the following vowel is more sonorous than i.

(195)

	Representation	Constraint	Repair	OK?
Input	· + · + [ka.ki][ta.ke]	LwProm		
		∗FinProm	↦ · + + · [ka.ki][ta.ke]	√
	· + + · [ka.ki][ta.ke]	WSP		
		∗iClash	↦ · · + · [ka.ki][ta.ke]	√
	· · + · [ka.ki][ta.ke]	NonHdCl		
		HdHdCl		
		∗HdProm	N/A	
Output	· · + · [ka.ki][ta.ke]			

The repair site for iClash is the ki syllable because the following syllable is more sonorous. remove-+ removes the prominence over the i from the autonomous grid. The resulting representation passes all higher and equal constraints and thus becomes the representation of choice. Even though this representation violates HdProm, it is the optimal representation given the hierarchy. There is no optionality here because the repair site is strictly determined.

The next derivation is for a case with optionality (194b).

(196)

	Representation	Constraint	Repair	OK?
Input	· + · + [i.ki][te.ti]	LwProm		
		∗FinProm	↦ · + + · [i.ki][te.ti]	√
	· + + · [i.ki][te.ti]	WSP		
Option i.		∗iClash	↦ · + · · [i.ki][te.ti]	√
	· + · · [i.ki][te.ti]	NonHdCl		
		HdHdCl		
		∗HdProm	N/A	
Output i.	· + · · [i.ki][te.ti]			
Option ii.		∗iClash	↦ · · + · [i.ki][te.ti]	√
	· · + · [i.ki][te.ti]	NonHdCl		
		HdHdCl		
		∗HdProm	N/A	
Output ii.	· · + · [i.ki][te.ti]			

This time the repair site for iClash is ambiguous; either one will do. The first option details the repair and result if the non-head is selected. The second option shows the results if the head is chosen as the repair site.

The case in (194c) does not involve an i-clash.

(197)

	Representation	Constraint	Repair	OK?
Input	· + · + [ka.maN][ta.ke]	LwProm		
		∗FinProm	↦ · + + · [ka.maN][ta.ke]	√
	· + + · [ka.maN][ta.ke]	WSP		
		iClash		
		∗NonHdCl	↦ · + · · [ka.maN][ta.ke]	√
	· + · · [ka.maN][ta.ke]	HdHdCl		
		∗HdProm	N/A	
Output	· + · · [ka.maN][ta.ke]			

The representation passes the iClash constraint, while it fails to pass NonHdCl. There is no ambiguity of site location for the NonHdCl repair because the non-head is always targeted.

The final example is for (194d), a case where the i syllable is followed by a heavy syllable.

(198)

	Representation	Constraint	Repair	OK?
Input	· + + · + [kaN.ti][mai][ta.cya]	LwProm		
		∗FinProm	↦ · + + + · [kaN.ti][mai][ta.cya]	√
	· + + + · [kaN.ti][mai][ta.cya]	WSP		
		∗iClash	↦ · · + + · [kaN.ti][mai][ta.cya]	√
	· · + + · [kaN.ti][mai][ta.cya]	∗NonHdCl	↦ · · + · · [kaN.ti][mai][ta.cya]	√
	· · + · · [kaN.ti][mai][ta.cya]	HdHdCl		
		∗HdProm	N/A	
Output	· · + · · [kaN.ti][mai][ta.cya]			

The iClash constraint locates the needed repair site as the ti syllable since the following syllable is more sonorous. The clash between mai and ta is also resolved by the designated repair for NonHdCl.

3.3.1.4 Heads with Extra-light Syllables

The foregoing has demonstrated how the original prominence projected on foot heads can be optimized to meet a set of prominence well-formedness constraints. Pichis Asheninca has one additional optimization strategy. There are two syllables, syi and ci, which repel prominence.[26]

Payne (1990) calls these extra-light syllables. This results in some intriguing patterns as exemplified by the forms in (199). The extra-light syllables are underlined.

(199)
- a. ó.ci.ti — 'dog'
- b. pí.ci.ci.ro — 'type of bird'
- c. pí.ci.rí.ne — 'your (type of) snake'
- d. nó.ci.ró.ne — 'my (type of) worm'
- e. ká.ci.tá.ke — 'he/she hurt'
- f. pí.syi.tá.ke — 'he/she swept'
- g. pí.syi.tá.ci.ri — 'broom (that which sweeps)'
- h. í.syi.ta.né.ta.tya — 'he has intestinal parasites'
- i. áa.weN.tá.roN.ci.tá.ci.ri — 'that which is medicine'

We can account for this display by adding another constraint to the hierarchy:[27]

(200) XProm: $* \overset{+}{\breve{\breve{\sigma}}}$

[26] Junko Itô (p.c.) notes a potential similarity between these extra-light syllables and voiceless high-vowels in Japanese. It is not inconceivable that the high vowel after the fricative or affricate devoices (or partially devoices). There would thus be a phonetic explanation for the lack of stress on such syllables: both intensity and pitch would necessarily be lower.

[27] Conceivably, one could try to account for the extra-lights by having the prominence process not assign a prominence for them. This approach runs into difficulties accounting for the absence of stress on the penult in a form like (199b). It also would constitute an exception to the generalization that all foot heads receive a prominence mark.

That is, avoid prominence on an extra-light syllable (syi or ci). Like FinProm, the XProm constraint does not involve clash. As a constraint dealing with intrafoot issues, its designated repair strategy is shift-+.

By Pāṇini's Theorem on Constraint-ranking (190), XProm must be ranked higher than iClash in the hierarchy since it is even more specific than iClash. As we have seen, WSP is also ranked higher than the clash constraints. XProm and WSP do not interact with each other because XProm only applies to light syllables. I will arbitrarily rank XProm above WSP as shown in the final ranking in (201).

(201) LwProm ≫ FinProm ≫ XProm ≫ WSP ≫ Clash ≫ HdProm

Several sample derivations are provided to show how this hierarchy achieves the results displayed in (199). For these derivations the cover constraint, Clash, will be used. First is (199c).

(202)

	Representation	Constraint	Repair	OK?
Input	· + · + [pi.ci][ri.ne]	LwProm		
		∗FinProm	↦ · + + · [pi.ci][ri.ne]	√
	· + + · [pi.ci][ri.ne]	∗XProm	↦ + · + · [pi.ci][ri.ne]	√
	+ · + · [pi.ci][ri.ne]	WSP		
		Clash		
		∗HdProm	N/A	
Output	+ · + · [pi.ci][ri.ne]			

As usual, FinProm shifts the prominence of the final foot onto the non-head. The resulting representation violates XProm since there is a prominence mark on an extra-light syllable. The shift-+ repair yields the representation to the right of XProm. This representation passes all higher and equal constraints, thus becoming the representation of choice.

The next derivation is for the near minimal pair found in (199b).

(203)

	Representation	Constraint	Repair	OK?
Input	· + · + [pi.ci][ci.ro]	LwProm		
		∗FinProm	↦ · + + · [pi.ci][ci.ro]	√
	· + + · [pi.ci][ci.ro]	∗XProm	↦ + · · · [pi.ci][ci.ro]	√
	+ · · · [pi.ci][ci.ro]	WSP		
		Clash		
		∗HdProm	N/A	
Output	+ · · · [pi.ci][ci.ro]			

This is similar to (202) except that there are two violations of XProm. For the left-most one, the shift-+ repair causes prominence to be marked on the non-head of the initial foot. shift-+ also applies within the final foot on the other violation site. Because prominence is already on the non-head, prominence is shifted "off the foot" resulting in a lack of prominence for the final foot.

The final derivation is for (199i).

(204)

	Representation	Constraint	Repair	OK?
Input	+ · + · + · + [aa][weN.ta][roN.ci][ta.ci]ri	LwProm		
		FinProm		
		∗XProm	↦ + · + + · + · [aa][weN.ta][roN.ci][ta.ci]ri	√
	+ · + + · + · [aa][weN.ta][roN.ci][ta.ci]ri	WSP		
		∗Clash	↦ + · + · · + · [aa][weN.ta][roN.ci][ta.ci]ri	√
	+ · + · · + · [aa][weN.ta][roN.ci][ta.ci]ri	∗HdProm	N/A	
Output	+ · + · · + · [aa][weN.ta][roN.ci][ta.ci]ri			

This example explains the stretch of two unstressed syllables, roN.ci, which actually constitute a foot.[28] While the head is assigned prominence, the resulting represen-

[28] Such a stretch of stressless syllables would be very difficult to explain via foot types under the

tation violates XProm. To repair this violation, the prominence is shifted to the roN syllable. This is now in clash with the preceding syllable and thus produces a violation of NonHdCl. The associated repair is to remove the prominence from roN.

There is a case where an extra-light syllable does bear stress:

(205) syí.ma 'fish'

This falls out directly from the analysis posited here as shown by the derivation in (206).

(206)

	Representation	Constraint	Repair	OK?
Input	$[\text{sy}\overset{\cdot}{\text{i}}.\text{m}\overset{+}{\text{a}}]$	LwProm		
		∗FinProm	$\mapsto [\text{sy}\overset{+}{\text{i}}.\text{m}\overset{\cdot}{\text{a}}]$	√
	$[\text{sy}\overset{+}{\text{i}}.\text{m}\overset{\cdot}{\text{a}}]$	∗XProm	$\mapsto [\text{sy}\overset{\cdot}{\text{i}}.\text{m}\overset{\cdot}{\text{a}}]$	∗
		WSP		
		Clash		
		∗HdProm	N/A	
Output	$[\text{sy}\overset{+}{\text{i}}.\text{m}\overset{\cdot}{\text{a}}]$			

The repair for FinProm produces a representation that violates XProm. The repair for XProm, though, will leave the word without any prominence markings; this violates LwProm. The repaired representation from FinProm thus becomes the best representation and is output.

3.3.1.5 Summary

By separating footing from prominence and positing the constraints in (201), the range of variability of stress placement in Pichis Asheninca is accounted for. By

usual assumption of a one-to-one correspondence between foot headship and stress. A form such as (154d) iN.kìN.ki.syi.re.tà.ko.tà.wa.ké.ri would be even more challenging due to its sequence of three stressless syllables ki.syi.re.

using an autonomous metrical grid which is related to but separate from footing, foot structure remains constant and simple. The complexities, which are limited to prominence and sonority issues, are all dealt with on the autonomous grid.

3.3.2 Prominence and Optionality

Before discussing issues of main stress assignment in Pichis Asheninca, it would be appropriate to discuss the implications of optional prominence placement for the parallel candidate set approach to optimization. Recall that this approach posits a potentially infinite set of candidates. The hierarchy of constraints selects the best candidate. How could optionality fit into such a model?

The first thing to consider is that the notion of an optional well-formedness constraint should not be admitted. What precisely would be denoted by such an idea? It would need to be a constraint asserting that a representation which meets its requirements is better formed than one that does not, yet this condition does not need to apply. This has the air of a contradiction: a representation that meets the constraint is 'better' than one that does not, yet a representation that does not meet the constraint can still be admitted as the 'best' formed output.

Note that this is not a problem for the notion of "Best Satisfaction". Having constraints which are optional would, in effect, produce several distinct hierarchies. The "Best Satisfaction" algorithm is always directly related to a particular instantiation of a hierarchy. The point of concern here is with the idea of how something that determines well-formedness could be conceived of as optional.

This problem does not rule out the possibility of optionality within the parallel candidate set approach, however. There are at least two other potential sources of optionality. These will be illustrated with the two cases of optionality in Pichis Asheninca stress repeated here as (207). The syllables undergoing optional stress assignment are underlined.

(207) a. i. i.kí.te.ti 'he is clean' (=184c)
ii. i.ki.té.ti

b. i. a.tí.ri.pá.yée.ni 'people' (=177b)
ii. a.tí.ri.pa.yée.ni

The optionality in (207a) could fall out straightforwardly within the parallel candidate set approach. Both (207a-i) and (207a-ii) are members of the candidate set. If there is a constraint such as iCLASH that prohibits stress clash between an i syllable and its following syllable, then both of these candidates will satisfy that constraint. As long as there is no constraint ranked lower than iCLASH in the hierarchy which eliminates one but not the other of these two candidates, then both will be selected as the "best". The partial tableau in (208) illustrates this.

(208)

Candidates	FINPROM	iCLASH
[i.ki][te.ti] (· + · +)	*!	
[i.ki][te.ti] (· + + ·)		*!
☞ [i.ki][te.ti] (· + · ·)		
☞ [i.ki][te.ti] (· · + ·)		

The best selection process will result in two choices. The usual optionality mechanisms will apply to select one or the other on a given instance. The point here is that the view employing candidate sets does have the ability to allow for optionality. Whenever the hierarchy is not able to select a unique candidate, the remaining candidates form the corpus of optional representations. While the intent and analytical practice is to always postulate enough constraints to select a unique candidate, this is not guaranteed in any way by the general theory of constraint ranking and best satisfaction.

The question that remains for Pichis Asheninca stress is whether it would be

possible or reasonable to assume that a constraint like iCLASH would indeed fail to be ranked above any constraint that would decide between (207a-i) and (207a-ii). If the ranking posited in (201) above were employed, then HDPROM would select (207a-i) over (207a-ii) since the former has only one violation of HDPROM while the latter has two. The required optionality would be eliminated.

The other potential source of optionality within the parallel candidate set approach would be to allow for indeterminate orderings of adjacent constraints in the hierarchy (as noted for Choctaw in Hung 1992). The idea is schematically represented in the constraint hierarchy of (209).

(209) $$\mathrm{A} \gg \mathrm{B} \gg \mathrm{C} \gg \left\{ \begin{array}{c} \mathrm{D} \gg \mathrm{E} \\ \mathrm{E} \gg \mathrm{D} \end{array} \right\} \gg \mathrm{F} \gg \mathrm{G}$$

The relative ranking of constraints D and E are indeterminate. Either D may dominate E or the other way around. If the ranking D ≫ E selects candidate O_i and the ranking E ≫ D selects candidate O_k, then the optionality of O_i versus O_k follows. (Again, the best candidate is selected relative to a particular instantiation of the constraint hierarchy.) The schematic tableaux in (210) and (211) illustrate this idea. Constraint D is ranked higher than E in (210), resulting in the selection of O_6.

(210)

Candidates	A	B	C	D	E	F	G
O_1	*!						
O_2			*!				
O_3				*!		*	
O_4			*!				
O_5		*!					
☞ O_6					*	*	

In (211), O_3 is chosen as the best because constraint E is ranked higher than D.

(211)

Candidates	A	B	C	E	D	F	G
O_1	*!						
O_2			*!				
☞ O_3					*	*	
O_4			*!				
O_5		*!					
O_6				*!		*	

Given the constraints posited here for Pichis Asheninca, this notion of indeterminate ranking of constraints works well for the alternation in (207b), repeated here as (212).

(212) i. a.tí.ri.<u>pá</u>.yée.ni ‘people’ (=177b)
ii. a.tí.ri.<u>pa</u>.yée.ni

If the relative ordering of Clash and HdProm is considered to be indeterminate, then the optionality is correctly predicted. Consider the two partial tableaux in (213) and (214).

(213)

Candidates	WSP	Clash	HdProm
· + · + + [a.ti][ri.pa][yee]ni		*!	
· + · · + ☞ [a.ti][ri.pa][yee]ni			*
· + · + · [a.ti][ri.pa][yee]ni	*!		*

When Clash is ranked higher than HdProm, (212ii) is selected.

(214)

Candidates	WSP	HdProm	Clash
· + · + + ☞ [a.ti][ri.pa][yee]ni			*
· + · · + [a.ti][ri.pa][yee]ni		*!	
· + · + · [a.ti][ri.pa][yee]ni	*!	*	

The other alternative (212i) is chosen when HdProm is ranked above Clash.

Unfortunately, there are two problems with employing this indeterminate ordering. First, the alternations in (207a) (repeated here as (215)) are not predicted.

(215) i. i.kí.te.ti ‘he is clean’ (=184c)
ii. i.ki.té.ti

This is demonstrated by the tableaux in (216) and (217).

(216)

Candidates	FinProm	Clash	HdProm
· + · + [i.ki][te.ti]	*!		
· + + · [i.ki][te.ti]		*!	*
· + · · ☞ [i.ki][te.ti]			*
· · + · [i.ki][te.ti]			**!

Having Clash ranked above HdProm yields (215i).

(217)

Candidates	FinProm	HdProm	Clash
· + · + [i.ki][te.ti]	*!		
· + + · [i.ki][te.ti]		*	*!
· + · · ☞ [i.ki][te.ti]		*	
· · + · [i.ki][te.ti]		**!	

The alternate ordering of Clash and HdProm also results in (215i) as the best candidate. Both orderings yield the same result here rather than allowing for optionality.

The second problem with employing the indeterminate ordering is exemplified by the forms in (218).

(218)	a.	kaN.ti.mái.ta.cya	'however'	(=189b)
	b.	oN.ki.tái.ta.má.na.ke	'in the morning'	(=189a)

These forms do not display any optionality, yet the indeterminate ordering hypothesis predicts that they will. The partial tableau in (219) shows the results for (218a) when Clash is ranked above HdProm.

(219)

Candidates	FinProm	Clash	HdProm
· + + · + [kaN.ti][mai][ta.cya]	*!	*	
· + + + · [kaN.ti][mai][ta.cya]		**!	*
· + + · · [kaN.ti][mai][ta.cya]		*!	*
· · + + · [kaN.ti][mai][ta.cya]		*!	**
· · + · · ☞ [kaN.ti][mai][ta.cya]			**

This selects the correct candidate. When HdProm is ranked over Clash, however, an unattested candidate is chosen as shown in (220).

(220)

Candidates	FinProm	HdProm	Clash
· + + · + [kaN.ti][mai][ta.cya]	*!		*
· + + + · [kaN.ti][mai][ta.cya]		*	**!
☞ · + + · · [kaN.ti][mai][ta.cya]		*	*
· · + + · [kaN.ti][mai][ta.cya]		**!	*
· · + · · [kaN.ti][mai][ta.cya]		**!	

Whether or not this poses a serious problem for this approach to optionality is a matter for further research. It is certainly the case that the set of ranked constraints posited here will not carry over unmodified to the parallel candidate set approach. A different set of constraints may fare better than the constraints required for the Constraint-Ranked Derivation approach given here. It is not immediately clear to me, however, what those constraints would be.

3.4 Main Stress

We now turn to address main stress assignment. Consider the following forms where secondary stress is marked by a grave accent (`) and primary stress by an acute accent (´). The syllable bearing primary stress is also underlined.

(221)
- a. sàa.sáa.ti — 'type of partridge'
- b. máa.ki.ri.ti — 'type of bee'
- c. ñàa.wyàa.ta.wá.ka.ri.ri — 'what he saw in a vision'
- d. nò.syi.ya.pì.ca.tàN.ta.ná.ka.ri.ri — 'I escaped from him' (=154c)
- e. no.tòN.ka.méN.to — 'my gun'
- f. iN.kìN.ki.syi.re.tà.ko.tà.wa.ké.ri — 'he thought about it for a while' (=154d)
- g. na.wì.sa.wè.ta.ná.ka — 'I went in vain'
- h. i.kàN.ta.syi.ta.rí.ra — 'he said it without thinking'

As Payne (1990:198) notes, primary stress falls on the heaviest stressed syllable within the final four syllables and, in the case of a tie, it falls on the rightmost stressed syllable. In addition, any stressed syllable to the right of the primary syllable loses its stress.

Payne also observes that the a nucleus actually behaves as more prominent than the other light syllables with respect to stress placement and behaves equally with e and o with respect to primary stress placement. We can make sense of this if we consider the relative sonority of these vowels as shown in (222).[29]

(222)

$$
\begin{array}{c}
\text{high sonority} \\
\text{main stress} \left\{ \begin{array}{c} \text{a} \\ \text{e o} \end{array} \right. \\
\left. \begin{array}{c} \text{e o} \\ \text{i} \end{array} \right\} \text{prominence well-formedness} \\
\text{low sonority}
\end{array}
$$

There is a distinct difference between the low a and the high i in sonority.[30] The mid vowels (e and o) pattern either with the low vowel or with the high vowel. One aspect of prominence well-formedness seems to focus on the low sonority of i. That is, prominence well-formedness is concerned with the lower end of the scale; it determines what the least acceptable prominence is. The low vowel a is clearly more sonorous than i, but the mid vowels are not distinct enough from i to behave in the same way that a does.

On the other hand, main stress assignment tends to focus more on the high sonority of a. That is, main stress is concerned with the higher end of the scale; main stress is applied to the most sonorous of the prominence markings. While the high vowel i is clearly less sonorous than a, the mid vowels are not distinct enough from a to behave like i does when it comes to main stress assignment.

[29] The relative sonority of these vowels is that given by Selkirk (1984a:112).

[30] Such a distinction also makes a phonological difference in Berber syllabification (Dell & Elmedlaoui 1985, 1988; see also Prince & Smolensky 1992, 1993b:11ff).

As mentioned in section 3.3.1.3, this distinction can be captured in traditional feature terms. The chart in (223) is repeated from that section.

(223) iCLASH — Main Stress (=188)

iCLASH	
`-low`	i,e,o
`+low`	a

Main Stress	
`+high`	i
`-high`	a,e,o

iCLASH focuses on the feature `low` while main stress focuses on `high`.

I will account for the stress in a way very similar to Hayes (1991). The End Rule process applies to the output from the prominence process and circumscribes the final colon. Within this colon, a separate, temporary, computational grid is projected for those syllables having prominence on the (permanent) autonomous metrical grid. This projection is based on sonority as given in (224).

(224) Main Stress Prominence Scale (Hayes 1991:250)
***: CVV
**: Ca, Co, Ce, CiN
*: Ci

The End Rule adds a mark on the metrical grid corresponding to the most prominent syllable on the prominence grid. The temporary prominence grid is then erased.

The output of this process is subject to the well-formedness constraint given in (225).

(225) MAINISFIN: Main prominence must be the final prominence in the word.

That is, the main prominence must be the last prominence at all levels. The repair strategy associated with MAINISFIN is REMOVE-+, the designated repair strategy for inter-foot constraints.

Sample derivations will be given for the forms in (226), repeated from earlier.

(226) a. ñàa.wyàa.ta.wá.ka.ri.ri 'what he saw in a vision' (=221c)

b. iN.kìN.ki.syi.re.tà.ko.tà.wa.ké.ri 'he thought about it for a while' (=154d)

The colon and the effect of MAINISFIN are exemplified by (226a).

(227)

	Representation	Constraint	Repair	OK?
Input	+ + · + · + [ñaa][wyaa][ta.wa][ka.ri]ri	LWPROM		
		FINPROM		
		XPROM		
		WSP		
		CLASH		
		HDPROM		
Output	+ + · + · + [ñaa][wyaa][ta.wa][ka.ri]ri			
EndRule	· + · + [ta.wa][ka.ri]ri * * *			
	‡ · + · + [ta.wa][ka.ri]ri	∗MAINISFIN	↦ ‡ · + · · [ta.wa][ka.ri]ri	√
Output	‡ + + · + · · [ñaa][wyaa][ta.wa][ka.ri]ri			
Result	ñàa.wyàa.ta.wá.ka.ri.ri			

Footing and prominence is straightforward. The end rule circumscribes the final colon, projects prominence on the temporary grid and marks the right-most prominent syllable for primary stress. The resulting representation is then submitted to the MAINISFIN constraint. The representation violates MAINISFIN so REMOVE-+ is invoked to produce the repaired representation. This is output with the final result as shown.

The derivation of (226b) is shown in (228).

(228)

	Representation	Constraint	Repair	OK?
Input	[iN.kiN][ki.syi][re.ta][ko.ta][wa.ke]ri	LwProm		
		FinProm		
		∗XProm	↦ [iN.kiN][ki.syi][re.ta] [ko.ta][wa.ke]ri	√
	[iN.kiN][ki.syi][re.ta][ko.ta][wa.ke]ri	WSP		
		∗Clash	↦ [iN.kiN][ki.syi][re.ta] [ko.ta][wa.ke]ri	√
	[iN.kiN][ki.syi][re.ta][ko.ta][wa.ke]ri	∗HdProm	N/A	
Output	[iN.kiN][ki.syi][re.ta][ko.ta][wa.ke]ri			
EndRule	[ko.ta][wa.ke]ri			
	[ko.ta][wa.ke]ri	MainIsFin		
Output	[iN.kiN][ki.syi][re.ta][ko.ta][wa.ke]ri			
Result	iN.kìN.ki.syi.re.tà.ko.tà.wa.ké.ri			

There is a violation of XProm which is repaired as shown. The kiN syllable is not subject to iClash because it has a coda. There is a violation of NonHdCl which also gets repaired. The result is a stressless three syllable sequence. The end rule circumscribes the final two feet, projects prominence and applies End Rule Right. The result is as shown.

3.5 Previous Accounts

Having laid out the proposed analysis, I will now directly address how it differs from two previous accounts of Pichis Asheninca stress. The original work is that of Payne (1990). Since it is couched in terms of the formalism of Halle & Vergnaud (1987a), I will refer to it as the HV account.

The other account is given in Hayes (1991:246–253). Hayes seeks to couch Payne's analysis within his Parametric Metrical Theory. His intent is to show how Parametric Metrical Theory can account for Pichis Asheninca stress, which he calls, "[t]he most complex prominence-based system known to me ..." Hayes (1991:246). I will call his analysis the PMT account.

Like Hayes, I have not fundamentally changed Payne's account. The manner in which the insights are expressed differs between the three accounts. The crucial distinction is that the one posited here avoids degenerate feet at all stages of the derivation. It also offers an explanation for the optionality exhibited in the stress system.

The framework of Halle & Vergnaud (1987a) allows Payne (1990) to account for the facts. Payne's analysis works for every form.

Unfortunately, such is not the case for the PMT account. As a rather minor point, Hayes (1991:253) notes that his account predicts that posttonic secondary stresses which are not in clash will surface. An example is given in (229), where the PMT account predicts (229a) while Payne reports (229b). The crucial distinction is underlined.

(229) a. máa.ki.<u>rì</u>.ti
      b. máa.ki.<u>ri</u>.ti

Hayes suggests that this discrepancy is small since the cues for posttonic secondary stresses can be subtle. As he rightly acknowledges, such posttonic secondary stresses are recorded for the Apurucayli dialect in Payne, Payne & Sanchez Santos (1982). Since Judith Payne did the analysis for the Apurucayali dialect (as indicated in the prologue of Payne, Payne & Sanchez Santos 1982:7), it seems extremely unlikely that she would hear the cues for one dialect, but not the other. This quibble could be overcome within the PMT account by positing a rule that deletes such posttonic

stresses. The deletion of such a stress entails the removal of the associated foot. The result would be a sequence of three unfooted syllables. The Constraint-Ranked Derivation account avoids such sequences by preserving footing and manipulating prominence marking on an autonomous grid.

More importantly, there is an area where the PMT account apparently fails to predict the facts correctly for extra-light syllables. If I understand the analysis correctly, when extra-light syllables are the head of a foot, the result is as in (230).

(230)	a.	(. x)		(x)	b.	(. x)		(x .)
		ŏ cĭ <t ĭ>	⟶	ŏ c <t ĭ>		kŏ syĭ <r ĭ>	⟶	kŏ syĭ <r ĭ>

If the extra-light is followed by a voiceless consonant, the i̲ nucleus of the extra-light deletes and forms a closed, heavy syllable. If the following consonant is voiced, however, the nucleus does not delete. The same kind of stress shift occurs, though, which I have represented as in (230b). Hayes (1991) is not explicit on this point, but I assume it must be something like this in order to account for the facts.

The PMT account posits the algorithm as in (231).

(231) Word Stress Algorithm (Hayes 1991:251)

a. Form a binary colon over the rightmost two feet. Labeling: End Rule Right, based on the prominence hierarchy of (232).
b. Apply the Prestress Destressing rule.
c. Eliminate a degenerate foot where it is not the only foot in the word.

The prominence hierarchy of (224) is repeated here.

(232) Main Stress Prominence Scale (Hayes 1991:250)

***: CVV
**: Ca, Co, Ce, CiN
*: Ci

The Prestress Destressing rule does the work of the clash constraints of section 3.3.1. Its details do not concern us for the examples at hand.

The empirical problem is for forms such as those in (233).

(233)	a.	pì.ci.rí.ne	'your (type of) snake'	(=199c)
	b.	nó.ci.ró.ne	'my (type of) worm'	(=199d)
	c.	ká.ci.tá.ke	'he/she hurt'	(=199e)
	d.	pí.syi.tá.ke	'he/she swept'	(=199f)

The PMT account would proceed as in (234) for (233a).

```
(234) a.  (          x)          b.  ——          c.  ( x          )
          ( x   . )  (x)                             ( x   . )    ∅
          pĭ  cĭ     rĭ <ne>                         pĭ  cĭ     rĭ <ne>
          *          *
```

The prominence grid and colon is constructed in (234a). The environment for Pre-stress Destressing is not met, so it does not apply as indicated in (234b). The degenerate foot is then removed with concomitant adjustment of the colon layer grid mark as shown in (234c). The predicted output is pí.ci.ri.ne with only one stress. The attested form, however, has two stresses: pì.ci.rí.ne. Thus, the PMT account does not correctly predict forms like those in (233).

The step which causes the analysis to go awry for these forms is the one which deletes the degenerate foot. If the degenerate foot were to remain, then the PMT account would indeed predict the correct two stresses in these forms. The degenerate foot deletion step appears to be questionable within Parametric Metrical Theory since it involves an apparent violation of the Continuous Column Constraint:

(235) Continuous Column Constraint (Hayes 1991:27):

> A grid containing a column with a mark on layer n + 1 and no mark on layer n is ill-formed. Phonological rules are blocked when they would create such a configuration.

When the degenerate foot is deleted in (234c), the colon grid mark is left stranded in what appears to be a violation of the Continuous Column Constraint. This constraint should then block the removal of the degenerate foot.

If this is correct, there are two difficulties, one empirical and one theoretical. The empirical difficulty is that the removal of these degenerate feet is essential for the PMT account to properly predict stress assignment for forms like those in (236).

(236)	a.	i.pí.co.ka	'he turned around'	(=154b-i)
	b.	pàa.ti.ká.ke.ri	'you stepped on him'	
	c.	kaN.ti.mái.ta.cya	'however'	(=189b)

If the degenerate foot is not deleted, then the PMT account predicts all of these to have an unattested stress on the penultimate syllable.

The theoretical difficulty is that degenerate feet would be not only allowed in the theory, but crucial for its empirical success. The motivation for step (c) in (231) was to avoid surface degenerate feet. This, then, seems to constitute a problem for the PMT account.

An advantage that the Constraint-Ranked Derivation analysis presented here has over the PMT account and the HV account is the lack of degenerate feet. Such feet are never built at all, while both the PMT account and the HV account crucially depend on degenerate feet at least at some point in the derivation. By ruling out degenerate feet, the Constraint-Ranked Derivation proposal maintains the one-to-one correlation between foot and minimal word. It also avoids the "exhaustiveness" problem whereby extrametricality must be overridden when extrametricality would remove the entire form from the purview of the operation at hand.[31] Finally, optionality is explained by the ambiguity of repair site location in the present account. Optionality must be stipulated in the other accounts.

Payne (1990) also discusses rapid speech rules and underlyingly stressed suffixes. Like Hayes (1991), I am not attempting to account for these. In addition, Payne (1990) notes that a bi-syllabic word consisting of an extra-light syllable followed by a

[31] As mentioned above in section 3.1, the solution to the latter problem is due to Prince & Smolensky (1992).

light syllable whose onset is voiceless results in a single syllable with a complex onset. That is, the nucleus of the extra-light deletes. I assume that this is a late phonetic effect: i.e. it is not a part of the phonology of stress assignment but rather a side effect of phonetic implementation.

3.6 Conclusion

The stress system of Pichis Asheninca presents challenges to the standard conception of the relation between footing and prominence on the metrical grid. By assuming that only heads of feet have prominence on the grid, certain forms in Pichis Asheninca require either degenerate feet or positing unwanted mixtures of foot types. By positing that the metrical grid is separate from, although related to, footing, these difficulties are overcome. Foot structure in Pichis Asheninca is kept simple. The complexities are all due to prominence well-formedness, not foot structure.

The proposal relies crucially on Constraint-Ranked Derivation. The ranked and violable nature of the well-formedness constraints allows for the range seen in the data. The notion that the constraints can pinpoint the locus of needed repair within a representation also allows an insightful explanation for optionality. Optionality is seen as a direct result of the application of a repair strategy to one of two possible repair sites.

Chapter 4

A Computational Implementation and its Implications for Theory and Analysis

4.1 Introduction

The previous three chapters introduced Constraint-Ranked Derivation and illustrated it with respect to stem well-formedness in Southeastern Tepehuan and stress in Pichis Asheninca. This chapter addresses issues relating to computationally implementing the Constraint-Ranked Derivation approach to optimization and the two analyses.

As mentioned in section 1.1, such a computational implementation (i) forces the analysis to be rigorous and precise, (ii) provides a means for thoroughly and exhaustively testing a larger set of data, and (iii) yields a tool for evaluating and doing further research. The sections that follow will discuss the results of the experiment of implementing Constraint-Ranked Derivation. The implementation successfully handled all of the data. Section 4.2 delineates some key results gleaned from the implementation's demands for rigor. Section 4.3 illustrates the tool by showing several instances of the output of the program. The next two sections provide detailed information regarding the data structures and key functions employed. These may be safely skipped by the non-computationally minded. The final section proposes some areas for future work.

4.2 Implications for Theory and Analysis

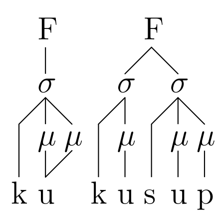
The exercise of computationally implementing the Constraint-Ranked Derivation approach led to several areas of improved precision. Several of these will be briefly discussed in what follows.

4.2.1 Prosodic Constituency

The approach to implementation taken here is to model the linguistic notions as conceived of by linguists. The implementation was not an attempt to view phonological processes from a purely computational point of view. This meant that the notions of prosodic constituency needed to be implemented in a linguistically meaningful fashion.

Consider the segment-mora-syllable-foot model assumed in this study as illustrated in (237).

(237)



The constituents, of course, are segments, moras (μ), syllables (σ), and feet (F). The lines drawn between these symbols represent a constituency relationship. A segment is a constituent of a syllable or a mora; a mora is a constituent of a syllable; a syllable is a constituent of a foot. At the same time, the constituency relation holds in the opposite direction: a foot has one or two syllables as sub-constituents; a syllable has segments and moras as sub-constituents; and a mora has one (or more) segments as a sub-constituent. There is a symmetric relation implied by such constituency lines.

This, of course, is no surprise. Nor is it necessarily surprising that the syllable and foot nodes are implicitly ordered with respect to each other. In (237), for example, the

$[\text{kuu}]_\sigma$ syllable linearly precedes the $[\text{ku}]_\sigma$ syllable which, in turn, precedes the $[\text{sup}]_\sigma$ syllable. In order to parse syllables efficiently into feet, the syllable data structures were implemented as a doubly linked list. That is, each syllable was represented as having a pointer or link to its immediately preceding and following syllable. From an autosegmental terminological point of view, these syllables are on the same "tier." Since constraints such as ClosedFtHd and Clash are foot oriented, the foot data structures were also implemented in a linked list.

The interesting point here is that while segments, syllables and feet clearly required such linkages, moras did not. There was no need to scan across a representation at the mora level as there was a need to scan at the segment, syllable, or foot level. No process, operation, or constraint required such a scanning. This raises the question as to whether or not the mora is truly of the same status in the prosodic hierarchy as the syllable and the foot. See Itô & Mester (1992:24–23, 32–34) for further discussion on this issue.

4.2.2 The Stem Binarity Constraint

The analysis of the truncation patterns of nominal stems in Southeastern Tepehuan included a Stem Binarity constraint. StemBin checks a representation to see if the stem portion is prosodically binary at some level of representation. If the input representation is more than binary, the constraint identifies one or more repair sites that will reduce the representation to a binary status. The repair is always remove-μ.

In order to implement this constraint, the identification of the repair sites needed to be made explicit. Chronologically, I did not actually figure out the needed method until I implemented this part of the program. It was clear from the data that binarity was always the result, but the stepwise procedure was not obvious.

The violations of StemBin fall into four categories as delineated in (238). One sample form is provided for each type.

(238)

	Type	Underlying	Surface	Gloss	
a.	F F σ	/too.si.ko.ri/	[tooš]$_F$[kor̃]$_F$	'pig'	(=33b)
b.	F F F	/naa.na.ka.sɨ.rV/	[naan]$_F$[ka.sɨr]$_F$	'scorpions'	(=62a)
c.	F F F σ	/taa.ta.kaa.rui.gV/	[taat]$_F$[ka.rui']$_F$	'chickens'	(=107b)
d.	F F F F	/haa.haa.vV.ka.ri-d/	[ha.haav]$_F$[ka.r̃i'ñ]$_F$	'his lungs'	(=113c)

The basic process is to first incorporate any final unfooted syllable into the final foot. Then the penultimate foot is removed for cases with three or more feet. Finally, for the (238d) case, the initial foot is removed by shortening the long vowel.

More precisely, the following sites are targeted in order to reduce these to two feet.

(239)
a. The mora of any final light syllable.
b. For cases (238b–d), the penultimate foot must be removed. This is accomplished by
   i. targeting the final mora of the head of this foot (if it is long); and
   ii. targeting the mora from any non-head syllable.
c. For the cases of (238d), the final mora of the head of the initial foot is targeted.

As a result of step (a), the final foot is headed by a closed syllable. In fact, in every case of types (238b–d) the final foot is either originally mono-syllabic or becomes mono-syllabic after the removal of the final mora.[1] This allows the preceding syllable to potentially become the non-head of the final foot. Step (239b-i) removes any extra mora in the preceding syllable to allow persistent footing to incorporate it into the final foot. The removal of the penultimate foot is completed by (239b-ii). The initial feet for case (238d) are a result of the bi-moraic reduplicative template. Therefore, they are always mono-syllabic, open syllables. By removing the final mora from the head, the syllable can become the non-head of the second foot. (The second foot in (238d) consists of a long syllable.)

[1] Why this should be so is an interesting, but unanswered, question.

4.2.3 Repair Sites and Domains

Another lesson from implementing the Southeastern Tepehuan analysis relates to the identification of repair sites and the domain of the repair. The ClosedFtHd constraint will be used as an example. Consider again the discussion in section 2.3.4.1 about the relative ranking of ClosedFtHd and StemBin. The key forms are those repeated in (240).

(240)	a.	/pii.pii.pi.ri/	⟶	[piip]$_{F}$[pir̃]$_{F}$	'chicks'	(=88d)
	b.	/sui.sui.ma.ri/	⟶	[suis]$_{F}$[mar̃]$_{F}$	'deer (pl)'	(=34a)

The correct ranking is for StemBin to be above ClosedFtHd as shown in (241) for (240a).

(241)

	Representation	Constraint	Repair	OK?
Input	[pii]$_{F}$[pii]$_{F}$[pi.ri]$_{F}$	ParseC		
		∗StemBin	[pii]$_{F}$[pi.pir]$_{F}$i	√
	[pii]$_{F}$[pi.pir]$_{F}$i	∗ClosedFtHd	↦ [piip]$_{F}$[pir]$_{F}$i	√
	[piip]$_{F}$[pir]$_{F}$i	∗ParseV	N/A	
Output	[piip]$_{F}$[pir]$_{F}$i			
Surface	[piip]$_{F}$[pir̃]$_{F}$			

If ClosedFtHd is ranked above StemBin, however, the derivation for (240a) would be as in (242).

(242)

	Representation	Constraint	Repair	OK?
Input	$[\text{pii}]_F[\text{pii}]_F[\text{pi.ri}]_F$	ParseC		
a.		*ClosedFtHd	↦ $[\text{pii}]_F[\text{pii}]_F[\text{pir}]_F\text{i}$	√
b.	$[\text{pii}]_F[\text{pii}]_F[\text{pir}]_F\text{i}$		↦ $[\text{pii}]_F[\text{piip}]_F\text{iri}$	*
c.			↦ $[\text{pii}]_F[\text{pi.pir}]_F\text{i}$	*
		*StemBin	$[\text{pii}]_F[\text{pi.pir}]_F\text{i}$	√
	$[\text{pii}]_F[\text{pi.pir}]_F\text{i}$	*ParseV	N/A	
Output	$[\text{pii}]_F[\text{pi.pir}]_F\text{i}$			
Surface	*$[\text{pii}]_F[\text{pi.pi}\tilde{\text{r}}]_F$			

The attempt to repair the initial foot (line c. under ClosedFtHd) is the step at issue. Since the attempted repair fails to close the initial foot, ClosedFtHd itself is violated. As originally conceived, the Constraint-Ranked Derivation algorithm stated that in such situations the ClosedFtHd constraint should be marked as inactive. A repair for a lower-ranked constraint would then be allowed to succeed in spite of the fact that the higher-ranked ClosedFtHd would be violated. If ClosedFtHd were still active, then the repaired representation for the lower-ranked constraint would fail when it was checked against ClosedFtHd. By marking ClosedFtHd as inactive, the lower-ranked constraint would be allowed to repair its violation.

This would work fine for the cases at hand. In the general case, however, it would be conceivable that the repair for a lower-ranked constraint could introduce a violation of a higher-ranked constraint that had been marked as inactive. In particular, using ClosedFtHd as an example, a foot whose head was closed by the repair associated with ClosedFtHd might become open by some lower-ranked constraint. This result, of course, should be a violation of the higher-ranked ClosedFtHd. If the ClosedFtHd constraint were no longer active, however, the violation would pass unnoticed. Implementing the Southeastern Tepehuan analysis brought this to my attention.

As mentioned in section 1.3.2, another way of viewing the situation is to notice that a constraint such as ClosedFtHd is concerned with the well-formedness of a particular domain, namely, the foot. The repair seeks to fix the domain with respect to the associated constraint. In the case where the repair fails to fix the violation, we can mark the particular domain as an exception to the constraint, rather than marking the entire constraint as inactive. This allows those domains which satisfy the constraint (either initially or as a result of repair) to continue to be checked for well-formedness throughout the rest of the optimization.

For this reason, the Constraint-Ranked Derivation algorithm (22) of section 1.3.2 has been modified to reflect the domain-oriented approach. The implementation follows suit. When a repair fails to fix a particular domain, the domain is labeled as an exception to the constraint. When the constraint is evaluated as a higher-ranked constraint, each domain is checked to see if it is an exception. If so, the domain is not checked.

4.3 Sample Output

This section provides some sample output from the program. The output consists of derivations very similar to the derivations given in chapters 2 and 3.

For stem optimization in Southeastern Tepehuan, the output consists of a footed input form and any optimizations that occur to it as it is passed through the four constraints. The first sample is in (243).

(243)
```
Input:                     [na.ka][s+.rV]
         ParseC
         StemBin
        *ClosedFtHd
Repaired representation: [na.ka][s+r](V)
        *ClosedFtHd
Repaired representation: [nak](a)[s+r](V)
Using:                     [nak](a)[s+r](V)
        *ParseV
Output:                    [nak](a)[s+r](V)
```

Footing is indicated by square brackets ([ ]) and an unparsed segment is enclosed in parentheses. The input form has two violations of ClosedFtHd: neither foot has a closed head. The rightmost foot is repaired first as shown, followed by the leftmost. The repaired representation violates ParseV.

The second sample is in (244).

(244)
```
Input:                     [baa][ba.nV]
         ParseC
         StemBin
        *ClosedFtHd
Repaired representation: [baa][ban](V)
        *ClosedFtHd
Repaired representation: [baab](anV)      *ParseC
Using:                     [baa][ban](V)
        *ParseV
Output:                    [baa][ban](V)
```

In this case, the repair of the leftmost foot produces a violation of the higher-ranked ParseC constraint. The output of the program reports which constraint the repair violates.[2]

The next example involves a violation of StemBin. The unfooted syllable $[\text{ri}]_\sigma$ is left unbracketed.[3]

[2]The derivations given in the text of this volume do not report the violated constraint due to formatting limitations.

[3]Segmental modifications such as palatalization are not modeled by the program.

(245)
```
Input:                      [tuu][tu.vuu]ri
          ParseC
         *StemBin
Repaired representation: [tuu][tu.vuur](i)
Using:                      [tuu][tu.vuur](i)
         *ClosedFtHd
Repaired representation: [tuut](u)[vuur](i)
Using:                      [tuut](u)[vuur](i)
         *ParseV
Output:                     [tuut](u)[vuur](i)
```

Recall that glides such as h may not be codas in Southeastern Tepehuan. This is why the repair to ClosedFtHd fails in (246).

(246)
```
Input:                      [voo]hi
          ParseC
          StemBin
         *ClosedFtHd
Repaired representation: [voo](hi)      *ClosedFtHd
Using:                      [voo]hi
          ParseV
Output:                     [voo]hi
```

Several other forms are now provided without comment.

(247)
```
Input:                      [sui][sui][ma.ri]
          ParseC
         *StemBin
Repaired representation: [sui][su.(i)mar](i)
Using:                      [sui][su.(i)mar](i)
         *ClosedFtHd
Repaired representation: [suis](ui)[mar](i)
Using:                      [suis](ui)[mar](i)
         *ParseV
Output:                     [suis](ui)[mar](i)
```

```
(248) Input:                        [pii][pii][pi.ri]
                ParseC
               *StemBin
      Repaired representation: [pii][pi.pir](i)
      Using:                        [pii][pi.pir](i)
               *ClosedFtHd
      Repaired representation: [piip](i)[pir](i)
      Using:                        [piip](i)[pir](i)
               *ParseV
      Output:                       [piip](i)[pir](i)

(249) Input:                        [tuu][tu.vuu][rid]
                ParseC
               *StemBin
      Repaired representation: [tuut](u)[vu.rid]
      Using:                        [tuut](u)[vu.rid]
                ClosedFtHd
               *ParseV
      Output:                       [tuut](u)[vu.rid]

(250) Input:                        [naa][na.ka][s+.rV]
                ParseC
               *StemBin
      Repaired representation: [naan](a)[ka.s+r](V)
      Using:                        [naan](a)[ka.s+r](V)
                ClosedFtHd
               *ParseV
      Output:                       [naan](a)[ka.s+r](V)

(251) Input:                        [taa][ta.kaa][rui]gV
                ParseC
               *StemBin
      Repaired representation: [taat](a)[ka.ruig](V)
      Using:                        [taat](a)[ka.ruig](V)
                ClosedFtHd
               *ParseV
      Output:                       [taat](a)[ka.ruig](V)
```

(252)
```
Input:                      [vaa][poo][tV.po][da.gV]
          ParseC
         *StemBin
Repaired representation: [va.poot](V)[po.dag](V)
Using:                      [va.poot](V)[po.dag](V)
          ClosedFtHd
         *ParseV
Output:                     [va.poot](V)[po.dag](V)
```

Turning to the Pichis Asheninca prominence optimization, the samples show how the prominence is initially projected and then optimized.

(253)
```
Word: oNkitaitamanake
Initial prominence:     .   +   +    .  +   .  +
                       [oN.ki][tai][ta.ma][na.ke]
 LwProm
*FinProm
Repaired prominence:    .   +   +    .  +   +  .
                       [oN.ki][tai][ta.ma][na.ke]
Using:                  .   +   +    .  +   +  .
                       [oN.ki][tai][ta.ma][na.ke]
 XProm
 WSP
*Clash (IHead)
Repaired prominence:    .   .   +    .  +   +  .
                       [oN.ki][tai][ta.ma][na.ke]
*Clash (NonHead)
Repaired prominence:    .   .   +    .  +   .  .
                       [oN.ki][tai][ta.ma][na.ke]
Using:                  .   .   +    .  +   .  .
                       [oN.ki][tai][ta.ma][na.ke]
*HdProm
Output prominence:      .   .   +    .  +   .  .
                       [oN.ki][tai][ta.ma][na.ke]
Final Prominence:                       +
                        .   .   +    .  +   .  .
                       [oN.ki][tai][ta.ma][na.ke]
   Output Results:  oNkitàitamánake
```

The initial prominence marking violates FinProm and is repaired. The clash with an i-head is fixed. The shift in final prominence produces a non-head clash which must be repaired. Final prominence is also projected. The output form includes stress marks.

The next sample involves two violations of XProm.

(254)
```
Word: piciciro
Initial prominence:       .  +   .  +
                        [pi.ci][ci.ro]
 LwProm
*FinProm
Repaired prominence:      .  +   +  .
                        [pi.ci][ci.ro]
Using:                    .  +   +  .
                        [pi.ci][ci.ro]
*XProm
Repaired prominence:      +  .   +  .
                        [pi.ci][ci.ro]
*XProm
Repaired prominence:      +  .   .  .
                        [pi.ci][ci.ro]
Using:                    +  .   .  .
                        [pi.ci][ci.ro]
 WSP
 Clash
*HdProm
Output prominence:        +  .   .  .
                        [pi.ci][ci.ro]
Final Prominence:         +
                          +  .   .  .
                        [pi.ci][ci.ro]
   Output Results:  píciciro
```

A case where XPROM must be violated is provided in (255).

(255)
```
Word: syima
Initial prominence:       .  +
                        [syi.ma]
 LwProm
*FinProm
Repaired prominence:      +  .
                        [syi.ma]
Using:                    +  .
                        [syi.ma]
*XProm
Repaired prominence:      .  .
                        [syi.ma]      *LwProm
Using:                    +  .
                        [syi.ma]
 WSP
 Clash
*HdProm
Output prominence:        +  .
                        [syi.ma]
Final Prominence:         +
                          +  .
                        [syi.ma]
   Output Results:  syíma
```

The assignment of final prominence to the most sonorous syllable is illustrated in (256).

(256)
```
Word: nosyiyapicataNtanakariri
Initial prominence:      .   +   .  +   .  +    .  +   .  +  .
                       [no.syi][ya.pi][ca.taN][ta.na][ka.ri]ri
 LwProm
 FinProm
*XProm
Repaired prominence:     +   .   .  +   .  +    .  +   .  +  .
                       [no.syi][ya.pi][ca.taN][ta.na][ka.ri]ri
Using:                   +   .   .  +   .  +    .  +   .  +  .
                       [no.syi][ya.pi][ca.taN][ta.na][ka.ri]ri
 WSP
 Clash
*HdProm
Output prominence:       +   .   .  +   .  +    .  +   .  +  .
                       [no.syi][ya.pi][ca.taN][ta.na][ka.ri]ri
Final Prominence:                                  +
                         +   .   .  +   .  +    .  +   .  .  .
                       [no.syi][ya.pi][ca.taN][ta.na][ka.ri]ri
   Output Results:  nòsyiyapìcatàNtanákariri
```

4.4 Data Structures

The next two sections discuss aspects of the implementation itself. The reader who is not interested in the programming side of the matter may safely skip these sections.

The program is written in the C programming language (Kernighan & Ritchie 1978) and has been compiled under MSDOS, System V Unix, and Sun OS. It employs several software libraries developed by SIL International including those employed in producing the programs documented in Black, Weber, Kuhl & Kuhl (1987); Weber, Black, & McConnel (1988); Weber, McConnel, Black & Buseman (1990); and Antworth (1990).

The program has almost ten thousand lines of code and comments. At the time

of writing it is still very much in an experimental state. Areas which remain to be improved or developed and tested include the user interface, feature tier manipulations, and cyclic or domain-oriented word building. The prosodic structures and functions required for dealing with the Constraint-Ranked Derivation approach to optimization for the Southeastern Tepehuan and Pichis Asheninca analyses are successfully implemented.

This section will describe some of the key data structures employed in the implementation. As mentioned above in section 4.2.1, the program models prosodic constituency structure. The nodes in the prosodic constituency are `word`, `foot`, `syllable`, `mora`, and `root_node`. In addition, the user defines the set of segments both in terms of the surface representation and in terms of featural content.[4] This set of `segment` structures delineates the valid set of segments and thus provides a potential means of determining structure preservation.

The data structures are presented in a bottom-up fashion, beginning with segments. Each segment in the set is stored in a `segment` structure as given in (257).

(257)
```
struct segment
    {
    char             *seg_str;  /* segment string */
    char             *seg_undr; /* underlying segment shape */
    int               seg_len;  /* length of segment string */
    struct root_node *seg_fs;   /* root node (feature hierarchy of the
                                   fully specified segment) */
    struct root_node *seg_us;   /* root node (feature hierarchy of the
                                   under-specified segment) */
    int               seg_mora; /* number of moras born by this segment */
    };
```

The first two strings distinguish the orthographic representation of the segment in surface and underlying form, respectively. Thus, a long high front vowel would have

[4]The latter can be represented in a feature geometry. Since the experiment documented here focused on prosodic issues, little use was made of the geometry.

`seg_str` point to the string `ii` and `seg_undr` point to the string `i`. The length of the segment is used in parsing input strings. The next two pointers refer to `root_node` structures which define the fully specified feature geometry and the (fully) underspecified feature geometry. The final declaration gives the number of moras (if any) born by the segment (typically one for short vowels and two for long vowels).

The root nodes have the structure in (258).

(258)
```
struct root_node
    {
    int               rn_son;    /* sonorancy   feature (+ or -) */
    int               rn_cons;   /* consonantal feature (+ or -) */
    struct dep_list  *rn_depl;   /* list of dependent nodes/features */
    struct mora      *rn_mora[MAX_MORAS];       /* pointer to mora(s) born
                                                    by this segment */
    struct syllable  *rn_syl;   /* pointer to syllable to which this
                                   segment belongs */
    struct segment   *rn_seg;   /* pointer to segment corresponding to
                                   this root node  */
    struct root_node *rn_left;  /* root_node to the left */
    struct root_node *rn_right; /* root_node to the right */
    };
```

Root nodes are stored in a doubly linked list to provide easy access to neighboring nodes. The `dep_list` is a pointer to the feature geometry of the node. The syllable and mora pointers represent the relation between the root node and its parent prosodic constituency. As implemented, every root node points to its syllable node even if it is moraic.

Moras are represented internally by the structure in (259).

(259)
```
struct mora
    {
    struct root_node_list *mora_rnl;    /* pointer to initial root node
                                           in mora */
    struct syllable       *mora_syl;    /* pointer to syllable containing
                                           the mora */
    };
```

The mora points to its syllable. It also points to its sub-constituent root nodes via a `root_node_list` structure. Such lists are used to point to one or more root nodes.

(260)
```
struct root_node_list
    {
    struct root_node      *rnl_rn;      /* pointer to root_node in list */
    struct root_node_list *rnl_next;    /* pointer to next element in */
                                        /* the list */
    };
```

The `root_node_list` structure allows for a variable number of root nodes to be associated with moras (and syllables).

Syllables are modeled by the structure in (261).

(261)
```
struct syllable
    {
    struct root_node_list *syl_rnl;     /* pointer to initial root node
                                           in the syllable */
    struct mora   *syl_mora[MAX_MORAS]; /* pointer to moras in the syl */
    struct foot           *syl_ft;      /* pointer to foot to which the */
                                        /* syllable is associated */
    struct strlist        *syl_cons;    /* list of constraints that this */
                                        /* syllable is allowed to violate*/
    int                    syl_wt;      /* weight of syllable */
    int                    syl_pr;      /* prominence projection */
    struct syllable       *syl_left;    /* syllable to left */
    struct syllable       *syl_right;   /* syllable to right */
    };
```

Like root nodes, syllables are implemented as a doubly linked list. A syllable points to its sub-constituent root nodes and moras and to its constituent foot. As discussed in section 4.2.3, domains may be marked for exceptionality to a constraint. The `syl_cons` pointer points to a list of strings containing the names of any such constraints. When a syllable is parsed, its weight is calculated and stored for quick reference. `syl_prom` is used for calculating stress prominence and has the values shown in (262).

(262)

Type	Value
UNSTRESSED	0
STRESSED	1
WORD_STRESS	2

The foot data structure is also a doubly linked list.

(263)
```
struct foot
    {
    struct syllable *ft_syl_left;      /* pointer to left  syl of foot */
    struct syllable *ft_syl_right;     /* pointer to right syl of foot */
    struct strlist  *ft_cons;          /* list of constraints that this*/
                                       /* foot is allowed to violate */
    struct foot     *ft_left;          /* foot to left */
    struct foot     *ft_right;         /* foot to right */
    };
```

It has pointers to a maximum of two syllables per foot. The `ft_cons` pointer lists the names of any constraints that the foot has violated.

The final prosodic structure is for the word.

(264)
```
struct word
    {
    struct root_node          *wd_rn;   /* initial root node in word */
    struct syllable           *wd_syl;  /* initial syllable in word */
    struct foot               *wd_ft;   /* initial foot in word */
    struct domain_repair_list *wd_dl;   /* current list of domains */
                                        /*  needing repair */
    struct foot      *wd_stress_ft;     /* foot     bearing word stress */
    struct syllable  *wd_stress_syl;    /* syllable bearing word stress */
    };
```

The `word` structure points to the initial root node, syllable, and foot in the word. It also marks the the foot and syllable bearing the word stress (the syllable could be determined from the foot if the headedness of the foot is known).

When a representation is evaluated by a constraint, the constraint may determine that one or more domains are in need of repair. This working list of domains is pointed to by the `wd_dl` pointer. The list of domains requiring repair is a singly linked list consisting of the structures given in (265).

(265)
```
struct domain_repair_list
        {
        union
          {                                /* domain to be repaired */
            struct syllable  *dr_syl;              /* pointer to syllable */
            struct foot      *dr_foot;             /* pointer to foot */
            struct word      *dr_word;             /* pointer to word */
          } dr_domain;
        int              dr_type;          /* type of domain */
        struct site_list *dr_sites;        /* list of repair sites */
        struct domain_repair_list *dr_next; /* pointer to next element in */
        };                                 /* the list */
```

The domain type is a syllable, foot, or word. The `dr_domain` union structure points to the domain itself within the representation. The `dr_sites` declaration points to a singly linked list of repair sites (266).

(266)
```
struct site_list
        {
        union
          {                                /* struct to repair */
            struct syllable  *si_syl;              /* pointer to syllable */
            struct mora      *si_mora;             /* pointer to mora */
          } si_rpar;
        int              si_type;          /* type of repair */
        struct site_list *si_next;         /* pointer to next element in */
        };                                 /* the list */
```

The type of repair is REMOVE-μ, ADD-μ, REMOVE-+, or SHIFT-+. The `si_rpar` union points to the structure to be repaired (which is either a mora or a syllable).

The constraints themselves are also stored in a singly linked list structure.

(267)
```
struct cons_list
        {
          char             *clst_name;       /* the external name used to   */
                                             /*      identify the constraint*/
          int             (*clst_func) ();   /* pointer to the function */
          int               clst_rpar;       /* designated repair strategy */
          int               clst_stat;       /* active/inactive status of */
                                             /*      the constraint */
          struct cons_list *clst_next;       /* link to next element in */
       };                                    /* the list */
```

The external name is how the user tells the program to employ the constraint via

the actual C-code function pointed to by `clst_func`. The designated repair strategy is either REMOVE-μ, ADD-μ, REMOVE-+, or SHIFT-+. The constraint can be marked as either active or inactive, thus allowing for lexical exceptions to mark particular constraints as inactive.

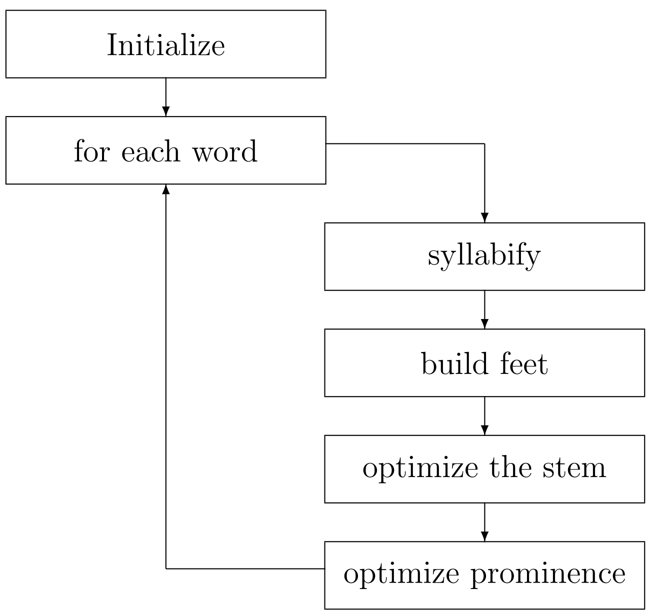
4.5 Functions

This section describes some of the key functions used in manipulating the data structures.

4.5.1 Overview

The essence of the overall function flow is given in (268).

(268)



Syllabification is based on sonority and is performed in a left-to-right sweep of the segments. Footing currently consists only of building iambs from left-to-right since both Southeastern Tepehuan and Pichis Asheninca employ iambic feet. Stem optimization consists of passing the output of footing through the Constraint-Ranked

Derivation algorithm described in more detail in section 4.5.2 below. Prominence optimization involves first projecting prominence on foot heads and then optimizing the projections via the Constraint-Ranked Derivation algorithm.

4.5.2 Constraint-Ranked Derivation

The most important function, of course, is Constraint-Ranked Derivation. The set of well-formedness constraints are hard-coded. Their ranking is determined by the user via a control file. For example, the control file for Southeastern Tepehuan would include the following lines:

(269)
```
\stem   ParseC
\stem   StemBin
\stem   ClosedFtHd
\stem   ParseV
```

The `\stem` code refers to stem well-formedness (`\prom` indicates prominence well-formedness). The constraints are placed in a singly linked list in the order in which they occur in the control file (see (267)).

The `crd` (Constraint-Ranked Derivation) function is passed a pointer to a word structure and a pointer to the list of constraints. It evaluates and modifies the representation stored in the word structure according to the hierarchy of constraints contained in the list. The potentially modified word structure is returned.

Each constraint is an integer function that returns a value of either `TRUE` or `FALSE`. When a constraint is called, it is given the word structure, an action flag, and the address of a domain list. The action flag is only effective for constraints which have an associated repair strategy. The flag has three values as delineated in (270).

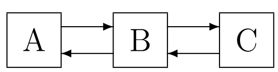
(270)

Value	Action
IDENTIFY_DOMAIN_REPAIRS	For every domain which fails the constraint, identify the domain and its needed repair sites.
VERIFY_REPAIR	For a specified domain, see if the attempted repair removed the violation.
REPORT_VIOLATION	Report any violation of the constraint.

The initial call to a constraint will build any needed list of domains to be repaired. The list of domains may be built in either a left-to-right or right-to-left fashion, depending on the constraint. The `crd` function processes the linked list in only one direction, but the list itself will be built either head first or tail first to provide the correct directionality.

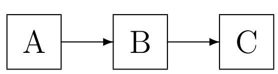
To see how this works, suppose there is a doubly linked list consisting of the three elements A, B, and C linked in that order as shown in (271).

(271)



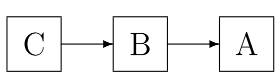
If the domain list is built from this list in a head-first fashion, then the domain list will be as in (272).

(272)



When this singly linked list is traversed, it will in effect traverse the original list in a left-to-right fashion (i.e. A, then B, then C). If the domain list is built in a tail-first manner, however, the domain list will be as shown in (273).

(273)



When this list is traversed, it will effectively traverse the original list in a right-to-left manner (i.e. C, then B, then A).

If the representation fails a constraint which has an associated repair strategy, then the `crd` function attempts to repair each violated domain in turn. The repair

is checked against the constraint (using the `VERIFY_REPAIR` action flag) to see if the repair succeeded in fixing the violation. If so, the repaired representation is checked against all higher-ranked constraints (which are invoked with the `REPORT_VIOLATION` action flag).

A copy of the representation is made before any repair is attempted. If the repair fails, the current word pointer is set to the copy. In this way, there is no need to try to "undo" the repair. Such an approach is conceivable. It could potentially be implemented by maintaining a stack structure which stored each step of the repair. The "undo" process would pop each step off the stack and invert it. The copy approach was used here since it was conceptually much simpler.

The essence of the C-code of the `crd` function is given below in (274).[5]

[5]Statements dealing with printing a trace of the evaluation have been removed for perspicuity.

(274)
```
struct word *crd(wp, cons_listp )
     struct word *wp;
     struct cons_list *cons_listp;
{
  struct cons_list *clp, *clp2;
  struct domain_repair_list *dlp;
  struct word *wp_hold;
  int i;
  int repair_attempted;
  int repair_failed;

                      /* initialize repair done flag */
  repair_attempted = FALSE;
                      /* check the representation (word) against */
                      /* the hierarchy of constraints */
  for (clp = cons_listp;
       clp != (struct cons_list *)NULL;
       clp = clp->clst_next)
    {
      if (clp->clst_stat == INACTIVE)
        {
          continue;             /* skip to next constraint */
        }
      if (!(*clp->clst_func)(wp, IDENTIFY_DOMAIN_REPAIRS, &(wp->wd_dl)))
        {              /* a constraint fails */
          if (wp->wd_dl == (struct domain_repair_list *)NULL)
            {         /* constraint does not have any repair strategy */
              if (!repair_attempted)
                {     /* no repair has been made; process is blocked */
                  return( (struct word *)NULL );
                }
                      /* a repair has been made; */
                      /* allow constraint to fail */
            }
          else
```

```
            {                   /* attempt to repair the representation */
                                /* init failure counter */
              repair_failed = 0;
                                /* do one domain at a time */
              for (dlp = wp->wd_dl, i=0;
                   dlp != (struct domain_repair_list *)NULL;
                   dlp = dlp->dr_next, i++)
                {
                                /* save the representation */
                  wp_hold = save_representation( wp );
                                /* repair violated domain */
                  repair_domain(wp, dlp);
                                /* check the domain to see if the repair */
                                /* fixed the original violation */
                  if (!(*clp->clst_func)(wp, VERIFY_REPAIR, &dlp))
                    {                   /* repair failed to fix site */
                      wp = restore_representation(wp, wp_hold, &dlp);
                      repair_failed++;
                      continue;     /* check next domain needing repair */
                    }
                                /* check resulting representation against
                                   all higher constraints */
                  for (clp2 = cons_listp;
                       clp2 != (struct cons_list *)NULL && clp2 != clp;
                       clp2 = clp2->clst_next)
                    {
                      if (!(*clp2->clst_func)(wp, REPORT_VIOLATION,
                                              &wp->wd_dl))
                        {               /* violated a higher constraint; */
                          wp = restore_representation(wp, wp_hold, &dlp);
                          repair_failed++;
                          break;    /* don't check any other constraints */
                        }
                    }
                                /* repaired rep becomes *the* rep to use */
                  if (wp != wp_hold)
                    free_word(wp_hold);
                  repair_attempted = TRUE;
                }               /* end of domain_repair_list loop */
            }                   /* end of attempt to repair */
        }                       /* end of a constraint fails */
    }                           /* end of loop to check constraints */
                        /* release domain_repair_list memory */
  wp->wd_dl = init_domain_repair_list(wp->wd_dl);
                        /* return optimized word */
  return( wp );

} /* end crd */
```

4.5.3 Constraints

The `crd` function invokes the various constraints. Two will be illustrated here. The first, ParseC, does not have any associated repair strategy. It initializes the list of domains if needed and then scans the root nodes to look for any unparsed consonants.

(275)
```
int parse_c( wp, action, drp )
     struct word *wp;
     int action;
     struct domain_repair_list **drp;
{
  struct root_node *rp;

                              /* initialize domain_repair_list if needed */
  if (action == IDENTIFY_DOMAIN_REPAIRS)
    *drp = init_domain_repair_list(*drp);

                /*  run through the linked list of root nodes and look */
                /*  for consonants */
  for (rp = wp->wd_rn;
       rp != (struct root_node *)NULL;
       rp = rp->rn_right)
    {
      if (rp->rn_cons == PLUS &&        /* found a consonant */
          rp->rn_syl == (struct syllable *)NULL)
        return(FALSE);                  /* no syllable */
    }
                /* no violation of constraint; return TRUE */
  return (TRUE);

}       /* end parse_c */
```

The other constraint is more interesting. StemBin does have an associated repair strategy and must therefore build both a list of violated domains and their associated repairs. The code is listed in (276). Discussion of the basic algorithm is in section 4.2.2.

(276)
```
int stem_bin( wp, action, drp )
     struct word *wp;
     int action;
     struct domain_repair_list **drp;
{
  struct mora *mp[MAX_SEGS];
  struct syllable *sp_last;
  struct foot *fp_last;
  struct stem *stp;
  int i, num_feet, num_sites;
                              /* initialize site list if needed */
  if (action == IDENTIFY_DOMAIN_REPAIRS)
    *drp = init_domain_repair_list(*drp);

                              /* count the number of feet while  */
                              /* finding the last foot */
  for (fp_last = wp->wd_ft, num_feet = 1;
       fp_last != (struct foot *)NULL &&
       fp_last->ft_right != (struct foot *)NULL;
       fp_last = fp_last->ft_right, num_feet++)
    ;

                              /* check for binarity */
  if (num_feet < 2)
    return( TRUE );           /* only one foot; is binary */

                              /* find last syllable */
  for (sp_last = wp->wd_syl;
       sp_last != (struct syllable *)NULL &&
       sp_last->syl_right != (struct syllable *)NULL;
       sp_last = sp_last->syl_right)
    ;

  if (num_feet == 2 && sp_last->syl_ft != (struct foot *)NULL)
        return( TRUE );       /* last syllable is footed, so stem */
                              /* meets binarity */
```

```
                              /* violates the constraint */
  stp = wp->wd_stem;
  switch (action)             /* perform appropriate action */
    {
    case VERIFY_REPAIR:        /* label stem as failing repair attempt */
      stp->st_cons = add_to_strlist(stp->st_cons, "StemBin");
      return(FALSE);           /* report failure */

    case REPORT_VIOLATION:  /* see if stem failed earlier repair attempt */
      if (member_strlist(stp->st_cons, "StemBin"))
        return(TRUE);          /* failed earlier repair attempt; is OK */
      else                     /* did not fail earlier repair attempt; */
        return(FALSE);         /* report failure */

    case IDENTIFY_DOMAIN_REPAIRS:   /* determine repair sites */
                                    /* add stem to domain repair list */
      *drp = add_to_domain_repair_list(STEM, (VOIDP *)stp, *drp,
                                 (struct site_list *)NULL,RIGHT_to_LEFT);
      num_sites = 0;          /* initialize site count */
                              /* final light syllable is always targeted */
      if (sp_last->syl_wt == LIGHT)
        mp[num_sites++] = sp_last->syl_mora[0];
      if (num_feet > 2)
        {                     /* head of penultimate foot must be light */
          if (fp_last->ft_left->ft_syl_right->syl_wt != LIGHT)
            mp[num_sites++] = fp_last->ft_left->ft_syl_right->syl_mora[1];
                             /* non-head of penultimate foot is truncated */
          if (fp_last->ft_left->ft_syl_left != (struct syllable *)NULL)
            mp[num_sites++] = fp_last->ft_left->ft_syl_left->syl_mora[0];
        }
      if (num_feet == 4)
        {                      /* head of initial foot is truncated */
          if (wp->wd_ft->ft_syl_right->syl_wt == HEAVY)
            mp[num_sites++] = wp->wd_ft->ft_syl_right->syl_mora[1];
        }
                               /* add moras to the site list */
      for (i = 0; i < num_sites; i++)
        (*drp)->dr_sites = add_to_site_list(REMOVE_MORA, (VOIDP *)mp[i],
                                      (*drp)->dr_sites, LEFT_to_RIGHT);
      break;
    }                    /* end of switch (action) */
                             /* if building repair sites, see if failed */
  if (action == IDENTIFY_DOMAIN_REPAIRS && domain_repair_needed(*drp))
    return(FALSE);
              /* no violation of constraint; return TRUE */
  return (TRUE);
} /* end stem_bin */
```

4.5.4 Repair of a Domain

The `crd` function will call the `repair_domain` function whenever it attempts to fix a domain. The repair function invokes a function that performs the appropriate repair operation for each site associated with the repair of the domain. The prominence oriented repairs (REMOVE-+ and SHIFT-+) are fairly straightforward: the prominence value is either removed from the syllable or else it is shifted to the non-head syllable of the foot (or removed if it is already on the non-head). The only data item affected by such repairs is the prominence integer value of a syllable structure `syl_pr` (see (261) and (262)).

The REMOVE-μ repair has much broader consequences. The removal of a mora may create the dissolution of a syllable or it may require the modification of foot structure. In the case of STEMBIN for Southeastern Tepehuan, an entire foot may need to be removed.

To implement this, the function `remove_mora` first removes the designated mora structure and all of its associated pointers. It then invokes `remove_syllable` which checks to see if the affected syllable needs to be removed. It also performs the essence of persistent syllabification.[6] This function in turn calls `remove_foot` to modify the structure of the affected foot. For example, a foot consisting of a single open long syllable will no longer be a legitimate foot when that syllable is shortened.

After `remove_mora` is called, a persistent footing function is invoked which incorporates unfooted syllables into feet whenever possible (degenerate feet are not allowed). At present, it assumes iambic feet.

The ADD-μ repair has yet to be implemented. It was not needed for either Southeastern Tepehuan or Pichis Asheninca.

[6] Admittedly, it would be better to have a persistent syllabification function be invoked whenever syllable structure was modified.

4.5.5 Save Representation

The final function to be discussed is the one which makes a copy of a representation. This conceptually simple routine proved to be quite a challenge. The difficulty was not so much making an in-memory copy of an existing word structure. The difficulty was in maintaining the list of domains and repair sites between the original and the copy.

Both the domains and the repair sites point to a particular instance of a foot, syllable, or mora in memory. The location of a particular foot, syllable or mora structure in the new copy is distinct from the original. A mapping between the old and new is required to ensure that the correct corresponding prosodic structure is referred to in the domain and repair site list of the copy.

The mapping was provided by a set of pointer arrays, one for each prosodic category. As each prosodic category was copied, the address of the old and new category were placed in the array index. When the domain list was built for the copy, its values were mapped from the appropriate index.

4.6 Future Work

While the experiment of implementing the Constraint-Ranked Derivation approach to optimization for Southeastern Tepehuan and Pichis Asheninca is successful, it still points to some areas for further work. One deals with the scope of the implementation; another with the means.

4.6.1 Non-Prosodic Optimizations

The Constraint-Ranked Derivation algorithm claims that any rule which is subject to well-formedness conditions can be modeled via the Constraint-Ranked Derivation approach to optimization. The examples explored in this study all dealt with

prosodic phenomena. Non-prosodic issues could also be explored. One such would anti-gemination and the OCP in Tonkawa (McCarthy 1986:223–225). Another could be the tradeoffs in some ATR Harmony systems noted by Archangeli & Pulleyblank (1993).

4.6.2 C++

As mentioned above, the program was implemented using the C programming language. C's `struct` specifier provides a flexible means to represent data structures and linked lists. Within C, however, there is no means to necessarily enforce the relations that exist between the various structures. Each structure is an independent entity and the relationship that exists between the structures must be maintained by the programmer. C's typecast mechanism would allow a programmer, for example, to have the `foot` structure point to a constraint list (`cons_list`) structure even though such a relation was never intended in the design of the data structures.

As I understand it, an object-oriented programming language such as C++ would provide the necessary enforcement of the intended relations between the data structures. Each prosodic category would be conceived of as an object and these objects could be arranged in a hierarchy. A mora object, for example, would be a subtype of a syllable object. Syllabification, footing, constraint evaluation and repair strategies would be operations on such objects. It would be interesting to implement the program in an object-oriented language such as C++. One aspect of this experiment would explore how well the rigor of object-oriented programming would allow the kinds of relations and representations that phonologists posit.

Chapter 5

Conclusion: Serial vs. Parallel Approaches to Optimization

The task addressed by this study was to provide analyses for two challenging sets of data, both of which revolve around footing and prosodic well-formedness: the Southeastern Tepehuan nominal stem truncations and stress in Pichis Asheninca. Building a computational implementation of the analyses was integral to this task. The notion of optimization with respect to a hierarchy of ranked and violable constraints as in Optimality Theory (Prince & Smolensky 1991, 1992, 1993a, 1993b, McCarthy 1993, McCarthy & Prince 1992, 1993) promised to be useful for gleaning insights into the data under consideration. The parallel candidate set approach espoused in Optimality Theory, however, posed serious obstacles for a computational implementation.

As a result, an alternative approach to optimization which was implementable was developed. This Constraint-Ranked Derivation approach to optimization allows the output of a process to be optimized per a hierarchy of ranked and violable constraints just as in the parallel candidate set approach of Optimality Theory. The Constraint-Ranked Derivation approach was shown to successfully analyze the complexities of the prosodic phonology and morphology of Southeastern Tepehuan and Pichis Asheninca.

The Constraint-Ranked Derivation approach requires a set of well-formedness constraints and a set of processes (or rules) much as traditionally conceived within Generative Phonology. Repair strategies can be associated with some constraints. These repair strategies are highly restricted; for example, prosodic repairs are either REMOVE-μ or ADD-μ and prominential repairs are either REMOVE-+ or SHIFT-+. Pro-

cesses may be subject to a hierarchy of ranked and violable constraints. A particular constraint may be included in the hierarchy of more than one process.

This modular approach to optimization allowed for succinct analyses of the Southeastern Tepehuan and Pichis Asheninca data. It also lent itself very well to computational implementation.

As mentioned in chapter 1, in order to produce an implementation under the parallel candidate set approach to optimization, one would need to determine precise algorithms for the generation function which produces the candidate sets. First, one would need to determine all sources of infinity and provide appropriate heuristics to limit the size of the set. Second, an algorithm to create all possible syllabic parsings of an arbitrary string would need to be developed. Third, one would need to create a procedure to produce all appropriate footings given a particular instantiation of the syllabification of an arbitrary string. Fourth, for each of these candidates, an algorithm would need to be developed to produce all possible feature linkings and delinkings. Fifth, all of these would need to interact with underspecified and partially specified representations, while maintaining the appropriate information that the set of constraints will need. Finally, morphological information such as stem and morpheme boundaries (including reduplicated material, McCarthy & Prince 1993:61ff) would need to be included in the representations in order to allow for constraints such as those in the ALIGN family (McCarthy & Prince 1993:32). (Such constraints deal with the interface between prosodic and morphological structure.)

The discussions about how Constraint-Ranked Derivation and the parallel candidate set approach could account for the Southeastern Tepehuan data brought out another consideration. The shortening of long vowels to meet Stem Binarity illustrated the potential need for the representation to include some way to tag surface short syllables which are underlyingly long. This and every other type of "globality of access" would need to be incorporated into the generation function to ensure that

it produced every necessary candidate.

The intriguing perturbations of stress in Pichis Asheninca provided another requirement for a full implementation of the parallel candidate set approach. The optionality that occurs in some of the clash resolutions would need to be implemented. Under the Constraint-Ranked Derivation approach, the constraints determine the locus of repair within a representation. Optionality is the result of the application of repair to one of two possible repair sites. The implementation simply needed to choose one of the repair sites.[1]

As mentioned in section 3.3.2, optionality might result from two sources under the parallel candidate set approach. First, although the intent and analytical practice is to always postulate enough constraints to select a unique candidate, that such a unique candidate will indeed be selected is not guaranteed in any way by the general theory of constraint ranking and best satisfaction. More than one candidate could conceivably survive the "Best Satisfaction" algorithm for a particular hierarchy of constraints. Such optionality would not require anything of the implementor; it is purely a function of the restrictiveness of the set of constraints in the hierarchy.

The second potential source of optionality is from the indeterminate ordering of two (or more) constraints (Hung 1992). To implement this source of optionality, one would simply need to evaluate the set of candidates on one (or all) of the various instantiations of the hierarchies.

All of the above would need to be addressed in order to produce an implementation of the parallel candidate set approach to optimization.

The Constraint-Ranked Derivation approach as developed in this study has demonstrated its viability as an approach to optimization. That it has been computationally implemented further demonstrates its rigor and precision.

[1] The actual implementation randomly chose one or the other.

References

Allen, W. Sidney (1973) *Accent and Rhythm*, Cambridge University Press, Cambridge.

Anderson, Stephen R. (1985) *Phonology in the Twentieth Century: Theories of Rules and Theories of Representations*, The University of Chicago Press, Chicago.

Antworth, Evan L. (1990) *PC-KIMMO: A Two-Level Processor for Morphological Analysis*, Occasional Publications in Academic Computing No. 16, Summer Institute of Linguistics, Dallas, Texas.

Archangeli, Diana B. (1984/88) *Underspecification in Yawelmani Phonology and Morphology*, Garland, New York. (Revision of 1984 Doctoral Dissertation, MIT, Cambridge, Massachusetts.)

Archangeli, Diana and Douglas Pulleyblank (1993) "Trade-off in ATR Harmony Systems," talk presented at Stanford University, Stanford, California, February 11, 1993.

Arnott, D.W. (1958) "The Classification of Verbs in Tiv," *Bulletin of the School of Oriental and African Studies*, University of London, 21:111–133.

Arnott, D.W. (1964) "Downstep in the Tiv Verbal System," *African Language Studies*, University of London, 5:34–51.

Black, Andy, David Weber, Fred Kuhl and Kathy Kuhl (1987) *Document Preparation Aids for Non-Major Languages*, Occasional Publications in Academic Computing No. 7, Summer Institute of Linguistics, Dallas, Texas.

Black, H. Andrew (1991) "The Phonology of the Velar Glide in Axininca Campa," *Phonology* 8:183–217.

Bliese, Loren F. (1981) *A Generative Grammar of Afar*, Summer Institute of Linguistics and University of Texas at Arlington Publications in Linguistics No. 65.

Cervantes Solís, Baudelio, Cornelio Ramírez Solís, Elizabeth R. Willett, and Thomas L. Willett (1991) *DICCIONARIO BILINGÜE PROVISIONAL En tepehuán y en español*, ms., Instituto Lingüístico de Verano, México, D.F.

Chomsky, Noam and Morris Halle (1968) *The Sound Pattern of English*, Harper and Row, New York.

Clements, George N. (1985) "The Geometry of Phonological Features," *Phonology Yearbook* 2:223–250.

Clements, George N. (1988) “Towards a Substantive Theory of Feature Specification,” *Proceedings of the North Eastern Linguistic Society* 18:79–93.

Clements, George N. and Samuel Jay Keyser (1983) *CV Phonology: A Generative Theory of the Syllable*, MIT Press, Cambridge, Massachusetts.

Dell, François and Mohamed Elmedlaoui (1985) “Syllabic Consonants and Syllabification in Imdlawn Tashlhiyt Berber,” *Journal of African Languages and Linguistics* 7:105–130.

Dell, François and Mohamed Elmedlaoui (1988) “Syllabic Consonants in Berber: Some New Evidence,” *Journal of African Languages and Linguistics* 10:1–17.

Goldsmith, John A. (1990) *Autosegmental and Metrical Phonology*, Basil Blackwell, Oxford.

Goldsmith, John (1991) “Phonology as an Intelligent System,” in Donna Jo Napoli and Judy Ann Kegl, eds., *Bridges between Psychology and Linguistics: A Swarthmore Festschrift for Lila Gleitman*, 247–267, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Goldsmith, John (1993) “Harmonic Phonology,” in John Goldsmith, ed., *The Last Phonological Rule: Reflections on Constraints and Derivations*, 21–60, University of Chicago, Illinois.

Grimes, Barbara F., Editor (1996) *Ethnologue: Languages of the World*, Summer Institute of Linguistics, Dallas, Texas.

Hale, Kenneth (1973) “Deep-surface Canonical Disparities in Relation to Analysis and Change: An Australian Example,” *Current Trends in Linguistics* 11:401–458.

Halle, Morris and Michael Kenstowicz (1991) “The Free Element Condition and Cyclic versus Noncyclic Stress,” *Linguistic Inquiry* 22:457–501.

Halle, Morris and Jean-Roger Vergnaud (1987a) “Stress and the Cycle,” *Linguistic Inquiry* 15:705–711.

Halle, Morris and Jean-Roger Vergnaud (1987b) *An Essay on Stress*, MIT Press, Cambridge, Massachusetts.

Hammond, Michael (1984) *Constraining Metrical Theory: A Modular Theory of Rhythm and Destressing*, Doctoral Dissertation, UCLA, Los Angeles, California; published 1988, Outstanding Dissertations in Linguistics series, Garland, New York.

Hammond, Michael (1987) “Hungarian Cola,” *Phonology Yearbook*, 4:267–269.

Hankamer, Jorge (1986) "Finite State Morphology and Left to Right Phonology," in Mary Dalrymple, Jeffrey Goldberg, Kristin Hanson, Michael Inman, Chris Piñon, and Stephen Wechsler, eds., *Proceedings of the West Coast Conference on Formal Linguistics* 5:41–52, The Stanford Linguistics Association, Stanford, California.

Hayes, Bruce (1981) *A Metrical Theory of Stress Rules*, Doctoral Dissertation, MIT, Cambridge, Massachusetts.

Hayes, Bruce (1984) "The Phonology of Rhythm in English," *Linguistic Inquiry* 15:33–74.

Hayes, Bruce (1985) "Iambic and Trochaic Rhythm in Stress Rules," in M. Niepokuj et al., eds., *Proceedings of the Thirteenth Meeting of the Berkeley Linguistics Society*, 429–446.

Hayes, Bruce (1987) "A Revised Parametric Metrical Theory," *Proceedings of the Northeastern Linguistic Society* 17:274–289.

Hayes, Bruce (1989) "Compensatory Lengthening in Moraic Phonology," *Linguistic Inquiry* 20:253–306.

Hayes, Bruce (1991) *Metrical Stress Theory: Principles and Case Studies*, ms., University of California, Los Angeles.

Hewitt, Mark and Alan Prince (1989) "OCP, Locality, and Linking: the N. Karanga Verb," in E.J. Fee and K. Hunt, eds., *Proceedings of the Eighth WCCFL*, CSLI, Stanford, California.

Hung, Henrietta J. (1992) "Relativized Suffixation in Choctaw: A Constraint-Based Analysis of the Verb Grade System," ms., Brandeis University.

Hyman, Larry (1985) *A Theory of Phonological Weight*, Foris, Dordrecht.

Inkelas, Sharon (1989/90) *Prosodic Constituency in the Lexicon*, Garland, New York. (Revision of 1989 Doctoral Dissertation, Stanford University, Stanford, California.)

Itô, Junko (1986) *Syllable Theory in Prosodic Structure*, Doctoral Dissertation, University of Massachusetts, Amherst.

Itô, Junko (1989) "A Prosodic Theory of Epenthesis," *Natural Language and Linguistic Theory* 7:217–259.

Itô, Junko (1990) "Prosodic Minimality in Japanese," in Michael Ziolkowski, Manuela Noske, and Karen Deaton, eds., *CLS 26-II: Papers from the Parasession on the Syllable in Phonetics and Phonology*, 213–239, University of Chicago, Illinois.

Itô, Junko, Yoshihisa Kitagawa, and R. Armin Mester (1992) "Prosodic Type Preservation in Japanese: Evidence from *zuuja-go*," Syntax Research Center publication SRC-92-05, University of California, Santa Cruz.

Itô, Junko and Armin Mester (1986) "The Phonology of Voicing in Japanese: Theoretical Consequences for Morphological Accessibility," *Linguistic Inquiry* 17:49–73.

Itô, Junko and Armin Mester (1989) "Optionality and Domain Selection," talk presented at University of California, Santa Cruz, November 3, 1989.

Itô, Junko and Armin Mester (1992) "Weak Layering and Word Binarity," LRC Working Paper 92-09, University of California, Santa Cruz.

Itô, Junko and Armin Mester (1993) "Licensed Segments and Safe Paths," revised version to appear in Carole Paradis, ed., *Constraints, Violations and Repairs in Phonology*, special issue of the *Canadian Journal of Linguistics*.

Kager, René (1993) "Alternatives to the Iambic-Trochaic Law," *Natural Language and Linguistic Theory* 11:381–432.

Kahn, Daniel (1976) *Syllable-based Generalizations in English Phonology*, Doctoral Dissertation, MIT, Cambridge, Massachusetts.

Karttunen, Lauri (1983) "KIMMO: a general morphological processor," *Texas Linguistic Forum* 22:165–186.

Karttunen, Lauri (1991) "Finite State Constraints," *International Conference on Current Issues in Computational Linguistics, June 10–14, 1991*, Universiti Sains Malaysia, Penang, Malaysia.

Karttunen, Lauri, Kimmo Koskenniemi and Ronald M. Kaplan (1987) "A Compiler for Two-Level Phonological Rules," ms., Xerox Palo Alto Research Center, Center for the Study of Language and Information, and Stanford University.

Kenstowicz, Michael and Charles Kisseberth, (1979) *Generative Phonology: Description and Theory*, Academic Press, Orlando, Florida.

Kernighan, Brian W. and Dennis M. Ritchie (1978) *The C Programming Language*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Kim, Jong Shil (1987) "Prosodic Analysis of Reduplication, Stress, and Vowel Deletion in Southeastern Tepehuan," ms., University of Texas, Austin.

Kiparsky, Paul (1982) "Lexical Phonology and Morphology," in I.-S. Yang, ed., *Linguistics in the Morning Calm*, 3–91, Hanshin, Seoul, Korea.

Koskenniemi, Kimmo (1983) *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*, Doctoral Dissertation, University of Helsinki, Finland.

Levin(Blevins), Juliette (1985) *A Metrical Theory of Syllabicity*, Doctoral Dissertation, MIT, Cambridge, Massachusetts.

Liberman, M. and A. Prince (1977) "On Stress and Linguistic Rhythm," *Linguistic Inquiry* 8:249–336.

Marantz, Alec (1982) "Re Reduplication," *Linguistic Inquiry* 13:435–482.

McCarthy, John (1979a) "On Stress and Syllabification," *Linguistic Inquiry* 10:443–466.

McCarthy, John (1979b) *Formal Problems in Semitic Phonology and Morphology*, Doctoral Dissertation, MIT, Cambridge, Massachusetts. Published 1985 by Garland Publishing, Inc., New York.

McCarthy, John (1986) "OCP Effects: Gemination and Antigemination," *Linguistic Inquiry* 17:207–263.

McCarthy, John (1988) "Feature Geometry and Dependency: A Review," *Phonetica*, 43:84–108.

McCarthy, John (1991) "Synchronic Rule Inversion," *Proceedings of BLS* 17:192–207.

McCarthy, John (1993) "A Case of Surface Constraint Violation," in Carole Paradis and Darlene La Charité, ed., *Constraint-based Theories in Multilinear Phonology*, special issue of the *Canadian Journal of Linguistics* 38:169–193.

McCarthy, John and Alan Prince (1986) "Prosodic Morphology," ms., University of Massachusetts, Amherst and Brandeis University.

McCarthy, John and Alan Prince (1988) "Quantitative Transfer in Reduplicative and Templatic Morphology," in Linguistic Society of Korea, ed., *Linguistics in the Morning Calm* 2, Hanshin Publishing Co., Seoul.

McCarthy, John and Alan S. Prince (1989) "Prosodic Morphology and Templatic Morphology," ms., University of Massachusetts, Amherst and Brandeis University.

McCarthy, John and Alan S. Prince (1990) "Foot and Word in Prosodic Morphology: the Arabic Broken Plural," *Natural Language and Linguistic Theory* 8:209–283.

McCarthy, John and Alan S. Prince (1991) "Prosodic Morphology," lecture notes and handouts from LSA Summer Institute, UCSC, Santa Cruz, California.

McCarthy, John and Alan S. Prince (1992) "Edge Alignment Constraints," talk presented at the University of California, Santa Cruz, December 11, 1992.

McCarthy, John and Alan S. Prince (1993) *Prosodic Morphology I: Constraint Interaction and Satisfaction*, ms., University of Massachusetts, Amherst and Rutgers University.

Mester, R. Armin (1986) *Studies in Tier Structure*, Doctoral Dissertation, University of Massachusetts, Amherst.

Mester, R. Armin (1990) "Patterns of Truncation," *Linguistic Inquiry* 21:478–485.

Mester, R. Armin (1994) "The Quantitative Trochee in Latin," *Natural Language and Linguistic Theory* 12:1–61 .

Mester, R. Armin and Junko Itô (1989) "Feature Predictability and Underspecification: Palatal Prosody in Japanese Mimetics," *Language* 65:258–293.

Moravcsik, Edith A. (1978) "Reduplicative Constructions," in J. Greenberg, ed., *Universals of Human Language, Vol. 3: Word Structure*, Stanford University Press, Stanford, California.

Munro Pamela and Peter John Benson (1973) "Reduplication and Rule Ordering in Luiseño," *International Journal of American Linguistics* 39:15–21.

Myers, Scott (1991) "Persistent Rules," *Linguistic Inquiry* 22:315–344.

Padgett, Jaye (1991) *Stricture in Feature Geometry*, Doctoral Dissertation, University of Massachusetts, Amherst.

Paradis, Carole (1988) "On Constraints and Repair Strategies," *The Linguistic Review* 6:71–97.

Payne, David L., Judith K. Payne, and Jorge Sanchez Santos (1982) *Morfología, fonología, y fonética del ashéninca del Apurucayali (campa-arawak preandino)*, Serie Lingüística Peruana, N^{o} 18, Instituto Lingüístico de Verano, Pucallpa, Perú.

Payne, Judith (1990) "Asheninca Stress Patterns," in Doris L. Payne, ed., *Amazonian Linguistics: Studies in Lowland South American Languages*, University of Texas Press, Austin.

Piggott, G.L. (1983) "Extrametricality and Ojibwa Stress," *McGill University Working Papers in Linguistics* 1:80–117, Montreal, Quebec.

Poser, Willaim J. (1990) "Evidence for Foot Structure in Japanese," *Language* 66:78–105.

Prince, Alan (1975) *The Phonology and Morphology of Tiberian Hebrew*, Doctoral Dissertation, MIT, Cambridge, Massachusetts.

Prince, Alan (1980) "A Metrical Theory for Estonian Quantity," *Linguistic Inquiry* 11:511–562.

Prince, Alan (1983) “Relating to the Grid,” *Linguistic Inquiry* 14:19–100.

Prince, Alan (1985) “Improving Tree Theory,” *Proceedings of BLS* 11:471–490.

Prince, Alan (1990) “Quantitative Consequences of Rhythmic Organization,” in Michael Ziolkowski, Manuela Noske, and Karen Deaton, eds., *CLS 26-II: Papers from the Parasession on the Syllable in Phonetics and Phonology*, 355–398, University of Chicago, Illinois.

Prince, Alan and Paul Smolensky (1991) “Connectionism and Harmony Theory in Linguistics,” lecture notes and handouts from the LSA Summer Institute at UCSC, Tech Report CU-CS-533-91, University of Colorado, Boulder.

Prince, Alan and Paul Smolensky (1992) “Optimality: Constraint Interaction in Generative Grammar,” paper given at the Eleventh *West Coast Conference on Formal Linguistics*.

Prince, Alan and Paul Smolensky (1993a) “Explanatory Typology through Optimality Theory,” talk presented at Stanford University, Stanford, California, February 11, 1993.

Prince, Alan and Paul Smolensky (1993b) *Optimality Theory*, ms., Rutgers University and University of Colorado, Boulder.

Pulleyblank, Douglas (1984) *Tone in Lexical Phonology*, Reidel, Dordrecht.

Sagey, Elizabeth (1986) *The Representation of Features and Relations in Nonlinear Phonology*, Doctoral Dissertation, MIT, Cambridge, Massachusetts.

Selkirk, Elizabeth O. (1982) “The Syllable,” in Harry van der Hulst and Norval Smith, eds., *The Structure of Phonological Representations*, Foris, Dordrecht.

Selkirk, Elizabeth O. (1984a) “On the Major Class Features and Syllable Theory,” in M. Aronoff and R.T. Oehrle, eds., *The Language Sound Structure: Studies in Phonology Presented to Morris Halle by His Teacher and Students*, MIT Press, Cambridge, Massachusetts.

Selkirk, Elizabeth O. (1984b) *Phonology and Syntax: the Relation between Sound and Structure*, MIT Press, Cambridge, Massachusetts.

Shaw, Patricia A. (1992) “Templatic Evidence for the Syllable Nucleus,” paper presented at NELS 23, University of Ottawa.

Shaw, Patricia A. (1993) “The Prosodic Constituency of Minor Syllables,” paper presented at WCCFL 12, University of California, Santa Cruz, April 4, 1993.

Smolensky, Paul (1986) “Information Processing in Dynamical Systems: Foundations of Harmony Theory,” in Rumelhart, David E., James L. McClelland, and the PDP Research Group, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Part 1*, MIT Press, Cambridge, Massachusetts.

Spring, Cari (1990a) "How many feet per language?" in Aaron L. Halpern, ed., *Proceedings of WCCFL* 9, CSLI, Stanford, 493–508.

Spring, Cari (1990b) *Implications of Axininca Campa for Prosodic Morphology and Reduplication*, Doctoral Dissertation, University of Arizona, Tucson.

Steriade, Donca (1982) *Greek Prosodies and the Nature of Syllabification*, Doctoral Dissertation, MIT, Cambridge, Massachusetts.

Steriade, Donca (1987) "Redundant Values," in *CLS 23-II: Papers from the Parasession on Autosegmental and Metrical Phonology*, 339–362, University of Chicago, Illinois.

Steriade, Donca (1988) "Greek Accent: A Case for Preserving Structure," *Linguistic Inquiry* 19:271–314.

Weber, David J., H. Andrew Black, and Stephen R. McConnel (1988) *AMPLE: A Tool for Exploring Morphology*, Occasional Publications in Academic Computing No. 12, Summer Institute of Linguistics, Dallas, Texas.

Weber, David J., Stephen R. McConnel, H. Andrew Black, and Alan Buseman (1990) *STAMP: A Tool for Dialect Adaptation*, Occasional Publications in Academic Computing No. 15, Summer Institute of Linguistics, Dallas, Texas.

Willett, Elizabeth (1981a) "Word Shortening in Southeastern Tepehuan," Masters Thesis, University of North Dakota, Grand Forks.

Willett, Elizabeth (1981b) "Noun Phrase Components in Southeastern Tepehuan," *Workpapers of the Summer Institute of Linguistics, University of North Dakota* 25:31–58, Huntington Beach, CA: Summer Institute of Linguistics.

Willett, Elizabeth (1982) "Reduplication and Accent in Southeastern Tepehuan," *International Journal of American Linguistics* 48:168–184.

Willett, Elizabeth (1985) "Palatalization in Southeastern Tepehuan," *International Journal of American Linguistics* 51:618–620.

Willett, Thomas Leslie (1991) *A Reference Grammar of Southeastern Tepehuan*, Summer Institute of Linguistics and University of Texas at Arlington Publications in Linguistics No. 100.

Winograd, Terry (1983) *Language as a Cognitive Process, Volume 1: Syntax*, Addison-Wesley, Reading, Massachusetts.

Yip, Moira (1988) "The Obligatory Contour Principle and Phonological Rules: A Loss of Identity," *Linguistic Inquiry* 19:65–100.

Zec, Draga (1988) *Sonority Constraints on Prosodic Structure*, Doctoral Dissertation, Stanford University, Stanford, California.