

COMPUTER SIMULATION OF SHAWNEE
HISTORICAL PHONOLOGY

CENTRE FOR NEWFOUNDLAND STUDIES

**TOTAL OF 10 PAGES ONLY
MAY BE XEROXED**

(Without Author's Permission)

TOWHID BIN MUZAFFAR



INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Computer Simulation of Shawnee Historical Phonology

by
Towhid Bin Muzaffar

A Thesis submitted to the School of Graduate Studies in partial fulfilment of the
requirements for the degree of Master of Arts

Department of Linguistics
Memorial University of Newfoundland

May 1997

St. John's

Newfoundland



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-25868-8

Canada

Abstract

In this study, an attempt has been made to use the computer program, PHONO, to develop a computer model which operates on the principle of the regularity of sound change. Surprising though it may seem, since this concept was first coined in modern linguistics, no standard computer program has been developed which could apply cross-linguistically to simulate and study diachronic change.

We have shown that the program may be used in the study of the Algonquian family of languages. We have developed a working model for the study of phonological change from the reconstructions of Proto-Algonquian to Shawnee. Using this Shawnee Model, we have simulated Shawnee data from reconstructed Proto-Algonquian words and illustrated a technique to collect or search for evidence for the reconstructed forms.

Given the nature and status of many languages of the world which are threatened with extinction, this technique may facilitate the preservation of valuable evidence for extinct pre-historic languages.

Table of Contents

Abstract	ii
List of Figures	v
Acknowledgments	v
List of Abbreviations	vi
Chapter One:	
Introduction	1
1.1 History of Proto-Algonquian	1
1.2 History of <i>A Computer Generated Proto-Algonquian Dictionary</i>	3
1.3 Refining a reconstruction	5
1.4 Methodology	6
1.5 Introduction to PHONO	7
1.6 The Shawnee Homeland	8
1.7 Previous works on Shawnee	10
Chapter Two:	
From PA to Shawnee	12
2.0 Introduction	12
2.1.1 Proto-Algonquian Phonology	12
2.1.2 Shawnee Phonology	13
2.2.1 Consonantal Clusters	13
2.2.2 Consonants	18
2.2.3 Vowels	18
2.2.4 Contractions	19
2.2.5 Other Rules	21
2.2.6 List of all the rules for PHONO	24
2.2.7 Rule Order	25
2.3 Reflexes	26
2.4 Morphology	34
2.4.1 Proto-Algonquian Verbal Morphology	36
2.4.2 Shawnee Verbal Morphology	37
Chapter Three:	
PHONO - Setting up the Shawnee Model	40

3.0	Introduction	40
3.1	Introduction to PHONO	40
3.2	The Alphabet File	41
3.2.1	The Alphabet File for the Shawnee Model	43
3.3	The Rule Makeup: phonological rules	44
3.3.1	The Rule Makeup File for the Shawnee Model	45
3.4	The Rule Order file	56
3.4.1	The Rule Order file for the Shawnee Model	57
3.5	Running PHONO	58
3.5.1	Interactive mode	58
3.5.2	Batch mode	59
3.5.3	Tracing the rules	60
3.5.4	Singleton	60
Chapter Four:		
	Running the Shawnee Model	62
4.0	Introduction	62
4.1	The Preliminary Search	62
4.1.1	Results and Discussion	69
4.1.2	Modifications to the Shawnee Model	70
4.2	The Shawnee data and the PA data	70
4.3	Method for comparing the data	72
Chapter Five:		
	Conclusion	73
5.1	Further development of PHONO	75
5.2	Other Applications of PHONO	76
5.2.1	Research Applications for Other Projects	76
5.2.2	Teaching the Regularity of Sound Change	76
	Bibliography	78
	Appendix 1	81
	Appendix 2	91
	Appendix 3	117

List of Figures

Key to Tribal Territories	9
---------------------------------	---

Acknowledgments

I would like to take this opportunity to thank the Department of Linguistics and the School of Graduate Studies of the Memorial University of Newfoundland for their financial support for my Master's program. I would like to express my gratitude to the faculty and the graduate students of the Linguistics Department for their constant support and help in various forms, such as lively discussions and exchange of ideas. My sincere appreciation goes to Dr. Kenneth R. Andrews and Rev. Ronald L. Chrisley for providing me with information from their field experience. I am particularly grateful to Rev. Chrisley for sending me a free copy of his book. My special thanks to Dr. Steven Lee Hartman of the University of Southern Illinois for his prompt e-mail responses to my tedious queries and for his permission to include the *User's Manual for PHONO* as an appendix. He gave me technical advice on using his program, PHONO, and developing the Shawnee Model. And, finally, to my supervisor, Dr. John Hewson, who has provided both financial and moral support beginning during my undergraduate years here at Memorial and continuing throughout my graduate program: thank you!

List of Abbreviations

AI	Animate Intransitive verb	PT	Pseudo Transitive Verb
an	Animate	PV	Preverb, Prenoun, also Root Initial
dim	Diminutive	pl	Plural
II	Inanimate Intransitive verb	Sh	Shawnee
inan	Inanimate	sg	Singular
MD	Multi-Dictionary	TA	Transitive Animate verb
NA	Animate Noun	TI	Transitive Inanimate verb
NI	Inanimate Noun	XP	Participle, including locatives and personal pronouns
N	Noun of uncertain gender		
obv	Obviative		
PA	Proto-Algonquian		

Chapter One:

Introduction

1.1 History of Proto-Algonquian

The study of Proto-Algonquian was put on a sound basis with Bloomfield's publication in 1925 of "On the sound-system of Central Algonquian" in the first issue of *Language*. In this paper, Bloomfield demonstrated that reconstruction is possible for unwritten languages, such as the Algonquian family of languages, just as it is possible for Indo-European languages, thereby showing that "the principle such as the regularity of phonetic change is not part of the specific tradition handed on to each new speaker of a given language, but either a universal trait of human speech or nothing at all, an error." (Bloomfield 1925: 130). This study of Bloomfield's was a continuation of a considerable number of comparative studies by several scholars, foremost among whom was Truman Michelson.

As early as 1914, Michelson had been doing Algonquian comparative work, and his papers "Algonquian Linguistic Miscellany" (1914), "Notes on Algonquian Languages" (1917), "Two Phonetic Shifts Occurring in Many Algonquian Languages" (1920), etc. gave a preliminary view of the linguistic relationships. Michelson's work continued after Bloomfield's 1925 article, and his forty page "Phonetic Shifts in Algonquian Languages" (1935) was described by Bloomfield as a "brilliant study of the divergent western languages" (1946: 85).

Considerable study has been done since, including checking and rechecking of the reconstructed forms by a whole range of Algonquian scholars, such as Charles Hockett (1957), Frank T. Siebert (1941, 1967, 1975), and Ives Goddard (1965, 1967a, 1967b, etc.). Hockett, in his 1957 article, published a list of all the Central Algonquian stems in /k-/. This was the first major step towards compiling a dictionary since Bloomfield (1925). Siebert (1967) is an important article on the original homeland of the Algonquian people. He uses natural history terms that have been determined for Proto-Algonquian and geographical distribution of species of birds, mammals, trees, and fish to work out the original homeland. Goddard (1965 and 1967a) proposes that there is only one subdivision in the Algonquian family in terms of genetic classification. Based on features shared by Abenaki, Delaware, Narragansett, Malecite-Passamaquoddy, and Micmac, he classifies this group as Eastern Algonquian. Goddard (1967b) is a major work in the reconstruction of Proto-Algonquian. In this paper, the paradigm of the independent indicative for Proto-Algonquian is proposed based on 16 languages from the Algonquian family.

All this study led to the publication of *A Proto-Algonquian Dictionary* by Aubin in 1975. This is a collection of Proto-Algonquian reconstructions of Bloomfield, Michelson, Hockett, and other scholars. It contains 2294 entries from twenty four sources by eleven scholars.

Finally, based on the regularity of sound change, computer methods were applied in the field of historical reconstruction of Proto-Algonquian by John Hewson. The result was the

publication of *A Computer-Generated Dictionary of Proto-Algonquian* in 1993, the production of which is discussed in Section 2 below.

The time depth of Proto-Algonquian (PA) may be considered as some 2000 - 3000 years.

The Algonquian family consists of the following languages (Goddard 1978: 70-1):

Plains:	Arapaho, Blackfoot, Cheyenne
Central:	Cree, Montagnais, Ojibway, Potawatomi, Menomini, Fox-Sauk-Kickapoo, *Miami-Illinois, Shawnee.
Eastern:	Micmac, Malecite-Passamaquoddy, “*Etchemin”, Eastern Abenaki, Western Abenaki, *Loup A, *Loup B, *Massachusett (or *Natick), *Narragansett, *Mohegan-Pequot, *Montauk, *Quiripi, *Unquachog, *Mahican, Delaware (Munsee, Unami), *Nanticoke, *Powhatan, *Carolina.

(The asterisk marks extinct languages.)

1.2 History of *A Computer Generated Proto-Algonquian Dictionary*

The computerized reconstruction project began in the early 1970s and culminated in the publication of the dictionary in 1993. Its objective was to use the computer to generate proto forms from the lexicons of daughter languages, and thus to produce proto-language dictionaries more simply and efficiently. The following is a summary of the four basic stages of the project.

Stage 1: The computer used the sound relationships or reflexes between each daughter language and the proto language to generate proto-forms. First it generated all the potential proto-forms from the known reflexes of the consonantal framework of each word. This procedure produced about 21 potential proto-projections for each word.

Stage 2: The program then sorted the massive list of proto-projections from all the languages being used in the reconstruction in alphabetical order within identical grammatical and semantic sets. Thus proto-projections that occurred more than once, generated from more than one language, were listed together. In other words, identical proto-projections from different languages, but having the same semantic and grammatical categories were printed out, and the remainder discarded. In this way, cognates along with the proto-projection from each cognate were isolated from the data.

Stage 3: The output was then edited and the vowels were added to these consonant frameworks to complete the process of reconstruction.

Stage 4: These were then carefully reviewed and after appropriate editing and adjustment, were included in the proto dictionary.

Cree, Fox, Menominee and Ojibway were the languages used as input data for the generation of the dictionary, and eleven other daughter languages and dialects were also cited in varying

degrees. They are listed below (the number of times each of them is cited is indicated in the bracket after each language or dialect): Micmac (24), Shawnee (22), Abenaki (5), Natick (4), Montagnais-Naskapi (3), Delaware (2), Moose Cree (1), Maliseet (1), Miami (1), Passamaquoddy (1), Penobscot (1).

1.3 Refining a reconstruction

The principle of refining a reconstruction is simple (Hock 1991: 607). Invariably in the reconstruction of a proto-language, the linguist or team of linguists is/are limited by the number of daughter languages used as evidence. In fact, the socio-linguist will argue that a considerable amount of valuable information is ignored when the various dialects of the daughter languages are not taken into account. Thus the principle of refining a reconstruction is to take a language or dialect that was not taken into consideration in the original work and compare it with the reconstructions. If a reconstruction is sound, there should normally be complete agreement between the reconstructed etymons and their reflexes in the new language being considered. Of course, for the cases in which there is incomplete agreement, the reconstruction may be “refined” in order to accommodate both the original daughter language(s) and the new evidence. This is the area in which the current research focuses. The total number of lexical items cited from Shawnee in the Proto-Algonquian dictionary, as indicated above, is 22. Miller (1959) has also cited about 150 Shawnee cognates in “An Outline of Shawnee Historical Phonology.” Thus considerable work still needs to be done in order to incorporate more Shawnee cognates into the dictionary. The ultimate objective

of the current study is to include as many Shawnee lexical items as possible in the Proto-Algonquian dictionary. Further, as well as the PA dictionary, Hewson produced a large list of reconstructed etymons (some 7000 items) which are based on evidence from only one language, currently referred to as “singletons”. This list was a product of reconstruction of proto-forms based entirely on comparison of formatives (see Hewson 1991, section I, “Algonkian word structure”). This procedure gives plausible etymons without corroborating evidence from another language. In these singletons, it is the formatives that are supported by the evidence of other languages, but the given combination of formatives, i.e. the total word, is not. It is anticipated that a number of these will also be supported by the Shawnee data; they can then be incorporated into the dictionary with evidence from two languages.

1.4 Methodology

The two primary objectives of the current study are:

- 1 to simulate Shawnee historical phonology using the computer program PHONO.
- 2 to use the results of this simulation to increase the evidence for the reconstructed forms in the Proto-Algonquian dictionary; to increase the dictionary by adding the singletons which are supported by the Shawnee data; and to refine those reconstructed forms which do not “fit” with the Shawnee data.

The first task undertaken was to establish the reflexes of PA in Shawnee. Once all the correspondences of the consonants, vowels and consonantal clusters and the glide-vowel

sequences were worked out, these rules of derivation of the Shawnee forms from the PA etymons were written in PHONO. We then supplied PHONO with a control set of PA forms. On running PHONO on the control set, we obtained the expected Shawnee forms. After reasonable success with a few control sets, we were ready to run the program on all the entries beginning with *p in the Proto-Algonquian dictionary and ultimately on the whole dictionary. Every time PHONO is run, the output requires appropriate adjustments; for example, the reflexes of the verbal paradigms of Proto-Algonquian in Shawnee have to be adjusted after PHONO provides us with the output. The final step is a thorough search of all existing dictionaries and word lists and other documents and perhaps even fieldwork to verify the entire output.

1.5 Introduction to PHONO

PHONO is a computer program written in Turbo Pascal by Dr. Steven Lee Hartman of the Department of Foreign Languages, Southern Illinois University. Dr. Hartman started working on developing a program that, when provided with:

- 1 a list of Latin words
 - 2 a complete list of sound changes
 - 3 the appropriate rule ordering or sequence of application of these sound changes
- would produce a list of Spanish words. He continued to refine this program so that it could be run on ANY language family.

The fundamental mode of operation of the program is based on the fact that it sorts the letters and makes specified changes to them using distinctive phonological features; hence its name PHONO.

In order to run PHONO on the Proto-Algonquian data, three steps are required:

- 1 an alphabet which includes all the phonemes of Proto-Algonquian and Shawnee, with all the distinctive feature of each phoneme specified.
- 2 a list of sound changes in terms of distinctive features.
- 3 a file containing the rule ordering of the sound changes; i.e. the order in which PHONO should apply the given sound changes.

The Shawnee Model was, thus, set up in PHONO with these basic steps. With this Model, reconstructed Proto-Algonquian forms were transformed to provide their expected reflexes in Shawnee.

1.6 The Shawnee Homeland

The homeland of the Shawnee in the second half of the eighteenth century was southern Ohio (Callender 1978: 622). This was also probably their homeland before the Europeans arrived in the continent. After several periods of dispersal, they ultimately settled in Oklahoma. The traditional tribal territory of the Shawnee is indicated in map 1 (*Handbook of North American Indians: Northeast* 15: ix).

According to Voegelin, Shawnee (in 1935) had three dialects: Absentee Shawnee, Eastern

Key to Tribal Territories

This map illustrates the historical territories of various Native American tribes in the eastern United States and southeastern Canada. Key features include:

- Geographical Features:** The Hudson River, James River, and the Appalachian Mountains are labeled.
- Tribal Territories:** Numerous tribes are identified, including the Algonquin, Iroquoian, Shawnee, and Delaware.
- Scale:** A scale bar at the bottom right indicates distances in miles (0 to 200) and kilometers (0 to 200).

Shawnee, and Loyal or Cherokee Shawnee (Voegelin 1935: 23). According to the *International Encyclopedia of Linguistics*, 200 speakers were reported in 1977 in central and northeastern Oklahoma. It states that most of these speakers were middle-aged or older. Ronald L. Chrisley, the author of *Introduction to the Shawnee Language*, estimates about 200 speakers of one of the dialects, about 15 speakers of the second and no speakers of the third dialect. Mr. Chrisley is a Methodist Minister and he admits that his approach in his book is that of a non-linguist. He estimates the number of the Shawnee people to be “a few thousand”. (Personal communications in April 1996). Kenneth R. Andrews who did the most recent field work at Shawnee, Oklahoma, for his PhD dissertation from University of South Carolina, 1994 entitled *Shawnee Grammar*, estimates the figure to be much lower than 200, somewhere around 50 to 60. (Personal communications in April 1996). He indicates that Tahlequah, North-East Oklahoma may be a promising locale for further field work.

Bloomfield classified Shawnee in the Sauk group; which includes Sauk, Fox and Kickapoo; but he called Shawnee “somewhat divergent” (Bloomfield 1925: 130). Later he placed Shawnee in a group by itself (1946: 85).

1.7 Previous works on Shawnee

Pentland and Wolfart, in the *Bibliography of Algonquian Linguistics*, list twenty five bibliographical entries for Shawnee. This list consists of two major works on Shawnee historical phonology. The first of these is “An Outline of Shawnee Historical Phonology” by

Miller (1959). This paper does not have a complete statement on semivowel combinations and syllable initial *e.

The second is Pentland's *Algonquian Historical Phonology* (1979). This is a comprehensive work consisting of seven chapters of which five are dedicated to individual languages, namely, Cree, Shawnee, Nawathinehena, Narragansett and Nanticoke. The last chapter is dedicated to Proto-Algonquian. The whole of chapter three is dedicated to the historical phonology of Shawnee. It cites PA forms and their reflexes in Shawnee. All the rules that are necessary to set up a Shawnee Model in PHONO are listed. This list includes several statements of irregular reflexes.

Chapter Two: From PA to Shawnee

2.0 Introduction

This chapter focuses on the historical phonology of Shawnee and very briefly deals with some morphological aspects of the transition from PA to Shawnee. In the first section, the phonemic inventories of PA and Shawnee are presented. In the second section, the PA reflexes of the consonants, vowels, consonantal clusters and some major vowel-glide sequences in Shawnee are cited. The next section begins with a list of reflexes of PA words in Shawnee. This section ends with some of the major issues in the historical phonology of Shawnee. These are mainly some of the irregular sound changes. The last section briefly addresses some morphological issues, namely some adjustments that have to be made to the PA data before running them in the Shawnee Model in PHONO.

2.1.1 Proto-Algonquian Phonology

The following is the phonemic inventory of PA; taken from Hewson (1993):

Consonants:

*p *t *č *k
 *θ *s *š *h
*m *n
 *l
*w *y

Vowels:

*i: *i *o: *o
*e: *e *a: *a

Consonantal Clusters:

Pre-Glottalized: *ʔt *ʔč *ʔs *ʔš *ʔθ *ʔl
Pre-Aspirated: *hp *hk *ht *hč *hs *hš *hθ *hl

Pre-Nasalized:	*mp	*nk	*nt	*nč	*ns	*nš	*nθ	*nl
Others:	*xp	*xk						
	*šp	*šk						
	*θp	*θk						
	*čp	*čk						
		*sk						

Rare and controversial clusters such as *št have been omitted.

2.1.2 Shawnee Phonology

The following is the phonemic inventory of Shawnee; taken from Voegelin (1935 and 1938):

Consonants:					Vowels:			
p	t	č	k	ʔ	i:	i	o:	o
m	n	š			e:	e	a:	a
		l						
w		y						

Consonantal Clusters:

Pre-Glottalized:	ʔp	ʔt	ʔk	ʔč	ʔš	ʔθ
Others:	šp		šk			

Whereas PA has a very elaborate system of consonantal clusters, (as many as 32, although two or three of these have very little supporting evidence) Shawnee has one of the simplest cluster systems in the entire family of Algonquian languages with only pre-glottalized clusters and šp and šk. This gives a total of 8 consonantal clusters in Shawnee.

2.2.1 Consonantal Clusters

The source of all the PA data is Hewson (1993) unless otherwise stated. The source of all

the Shawnee data is Voegelin (1938) unless otherwise stated. In citing the PA data, the orthography used is the same as that in Hewson (1993), alternatively called the Multi-Dictionary (MD) or simply the dictionary. The number associated with each word is the number from the dictionary and is labeled “MD”. The Shawnee data are transcribed as in Voegelin (1938). For many Shawnee words, the *Shawnee stems and the Jacob P. Dunn Miami Dictionary* Voegelin (1938), has several English glosses. For the sake of conciseness, only one English gloss (sometimes more) was selected for such words. At least one PA word and its reflex in Shawnee is cited for each cluster. For each rule, first the rule is cited in notation form; in the next line, first the PA word is cited followed by the Shawnee reflex, the Shawnee gloss and the MD number of the PA word. The clusters that do not undergo any sound change are simply listed, without any citation of PA words and their Shawnee reflexes. This is because section 2.3 has a list of reflexes. This format is also used throughout section 2.2.

Pre-aspirated clusters: All the PA pre-aspirated clusters replace aspiration by glottalization in Shawnee.

*hs > ʔθ			
*neʃihsa	nisiʔθa	my uncle	2328
*nemihsa	nimiʔθa	my older sister	2266
*hl > ʔθ			
*le:hle:weni	leʔθeewe	soul, breath	1428
*ht > ʔt			
*wanihta:wa	niwanʔto	I missed it	3550

*nehtawakayi	nʔtawaka	my ear	(Aubin 1975: 31)
*hk > ʔk			
*axkehkwā	haʔkoʔkwa	bucket, kettle, vessel	0475
*e:mehkwa:n-	heemʔkwa, -aanaki	spoon, -s	0602
*hš > ʔš			
*ahši:mini	haʔšiimi	papaw	0162
*ki:šwawihšinwa	kiišoʔšinwa	he lies warmly covered	1286
*ša:hša:k-	šaʔšaak- (-aw-, -anaw-)	stripped, naked	3017
*hθ > ʔθ			
*meta:hθwi	metaʔθwi	ten	1891
*no:hθa	noʔθa, noʔθaki	my father-s	2414
*hp > ʔp			
*a:hpawe:wa	haʔpowe	he dreams	0015
(The unstressed “a” is raised to “o” in Sh under the influence of the following “w”).			
*weta:hpantamwa	nootaʔpata	I sucked it	3851
*wi:hpe:me:wa	niwiʔpeema	I sleep with him	3935
*hč > ʔč			
*či:hči:peška:wa	čiʔčiipeška	he trembles	(Pentland 1979: 136)

Pre-glottalized clusters: The PA pre-glottalized clusters do not undergo any change except

“ʔs” and “ʔl”. Both of these change to “ʔθ”.

*ʔs > ʔθ			
*name:ʔsa	nameʔθa, nameʔθaki	fish, many fish	2142
*wi:ʔsakanwi	wiʔθakanwi	it is bitter	4012
*wi:θeʔsi	wiileʔθi, wiileʔθa	head hair-s, his head hair	4050
*ʔl > ʔθ			
*pemiʔle:wa	pemʔθe	he goes by	2627

The clusters that remain unchanged are:

*ʔt > ʔt
 *ʔč > ʔč
 *ʔš > ʔš
 *ʔθ > ʔθ

Pre-nasalized clusters: The nasal segment is lost in all the PA pre-nasalized clusters except the *nɭ cluster which changes to “n”.

*mp > p			
*wempi-	hop-	up	3719
*wetempi	hotepi (nitepi)	his brain (my brain)	3886
*nč > č			
*neθenčyi	nileči	my finger	2347
*wa:pančike:wa	waapačike	he looks on	3427
*wenči-	hoči-	from, away from	3726
*nk > k			
*penkwi	pekwi	dust, ashes, worn out, sore.	2654
*wa:panke	waapake	tomorrow	3430
*nš > š			
*neški:nšekwi	nʔškiišekwi, nʔškiišeko	my eye, -s	2332
*nt > t			
*pi:nte:wi	piit-	in, in something, inside, hole	2743
*wi:ntamwa	niwiita	I named it	4005
*nɭ > n			
*wi:nle:wa	niwiina	I named him	3997
*ns > θ			
*ameθkw(V:ns)a	ameʔkooθa	small beaver	(Pentland 1979: 143)
*ši:ʔši:p(V:ns)a	šiʔšiipeeθa	small duck	(Pentland 1979: 143)
*nθ > n			
*ma:nθehsi	maaneʔθi	flint	1496
*nepi:nθaha:ni	nipiinha	I stuck it in	(Pentland 1979: 141)

*neθkenθeki	n(?)kineki	my forehead	(Pentland 1979: 141)
-------------	------------	-------------	----------------------

Other clusters: Except *šp and *šk, all the other clusters become pre-glottalized.

*xp > ʔp			
*axpapiwa	haʔpapi	sitting on something	0508
*wexpenya	hoʔpenya, hoʔponyeeki	prewhite sweet potato, -es	3912

*xk > ʔk			
*axkehkwa	haʔkoʔkwa	bucket, kettle, vessel	0475

*θp > ʔp			
*eθpwekwatwi	ʔpokwanwi	it tastes in some way	0823

*θk > ʔk			
*ameθkwa	hameʔkwa	beaver	0261
*po:θkama	poʔkama	peach	2823
*weθkani	hoʔkani, hoʔkanali	bone, his bones	3898

*čp > ʔp			
*kwečpenwa	kwpenwa	he dreads something	(Pentland 1979: 134)

*čk > ʔk			
*walakačkwali	holakaʔkooli	his palate	(Pentland 1979: 138)
*nepačkyama:wa	nipkama	I bite off a piece of him	(Pentland 1979: 138)
*kečkyatwi	kkatwi	it is old	(Pentland 1979: 138)

*sk > šk			
*meskw(y)e:wi	mškwaaawi	it is red	(Pentland 1979: 138)
*wenči-eskatwi	očiškatwi	it (water) dries up	(Pentland 1979: 138)

The following clusters remain unchanged:

*šp > šp
*šk > šk

2.2.2 Consonants

The only PA consonants that undergo change are *θ and *s.

*θ > l			
*neθenčyi	nileči, nilečali	my finger, -s	2347
*θeʔθe:ma:wa	laʔθeema	tobacco	3394
*wi:θani	wiilani	the tongue, his tongue	4049

*s > θ			
*ki:šesamwa	nikiišeθa	I cooked	1222
*si:pi:wi	θiipi, θiipiiwali	river-s	3001

All the other PA consonants remain unchanged:

*p	>	p
*t	>	t
*k	>	k
*č	>	č
*š	>	š
*h	>	h
*m	>	m
*n	>	n
*l	>	l
*w	>	w
*y	>	y

2.2.3 Vowels

*e	>	i / #k_		
*kešyete:wi		kišite	it is hot	1109

*i:	>	i / _y		
*wi:yaka:kamyi		wiyakaakami	dirty water	(Pentland 1979: 161)
*wi:yaweysi		wiyawʔθi	meat	(Pentland 1979: 161)

This rule applies “at least in initial syllables” (Pentland 1979: 161).

Elsewhere, all PA vowels remain unchanged in Shawnee.

Long:

*i:	>	i:
*e:	>	e:
*a:	>	a:
*o:	>	o:

Short:

*i	>	i
*e	>	e
*a	>	a
*o	>	o

2.2.4 Contractions

Thirteen PA glide plus vowel sequences and their reflexes in Shawnee are cited below.

*yi	>	i / {-syllabic}_		
*anyikwa		hanikwa	squirrel	0276
*nepyi		nepi	water	2292
*yi:	>	i: / {-syllabic}_		
*nyi:šwi		niišwi	two, pair, double, together	2462
*ye	>	i {-syllabic}_		
*wasye?θanwi		wθi?θa	nest	3558
*ye:	>	e: / {č, š}_		
*no:nšye:-		no:še	have offspring, young	2420
*wečye:pihki		hočeepi?ki	medicine	3652
*wečye:wa		hoče, hočeeki	fly, flies	3655
*ya	>	ya / _#		
*wexpenya		ho?penya, ho?ponyeeki	prewhite sweet potato, -es	3912
(*ya > *ye:, in PA, when between two consonants.)				
*ya:	>	a: / {č, š, w}_		
*čya:ki-		čaak-, čaak(y)-	group numbering all	0593
*wesa:wya:wi		hoθaawa	yellow	3806
*wi	>	wi		
*ka:wiya		kaawi, kaawiiki	briar, -s	0872
*le:kawi		leekawi	sand	1430
*wa:panwi		waapanwi	it is morning	3437

*twi > ti	(except in II verbs)		
*nekwetwi	nekoti, nekotw-	one, alone	2258
*wi:hpe:ntwiwaki	niwi?peetipe	we sleep together	3937
*wi: > wi:			
*wi:kopyi	wiikopi, wiikopiwali	smooth bark, -s	3948
*wi:ni	wiini	kernel, marrow	3980
*we > o / {C, #}_			
*nekwetwi	nekoti, nekotw-	one, alone	2258
*wetempi	hotepi (nitepi)	his brain (my brain)	3886
*eθpwekwatwi	?pokwanwi	it tastes in some way	0823
*we: > we:			
*apwe:wa	hapweewa	roasting, barbecuing	0301
*a:hpawe:wa	ha?powe	he dreams	0015
*wa > o / #_			
*walahakaya	hol?haka	scale of fish or reptile	3530

This rule does not apply to the root *wan- (lose):

*wanehše:wa wane?še he hears wrong (Pentland 1979: 160)

*wa > wa			
*ameθkwa	hame?kwa	beaver	0261
*apwe:wa	hapweewa	roasting, barbecuing	0301
*pepikwani	pepikwa	musical instrument	2657
*wa: > wa:			
*takwa:kiwi	takwaaki	it is autumn	3203
*wa:pame:wa	niwaapama	I look at him	3423
*wa:panwi	waapanwi	it is morning	3437

Note: In the initial syllable, the regular reflex of *y followed by a long vowel produces an extra syllable as follows:

*mye:wi	miyeewi	road, path	2063
*nye:wi	niyeewi	four	2444

*nya:ʔanwi	niyaalanwi	five	2439
*pye.ta:wa	nipiyeeto	I bring it	2868
*pyi:mahamwa	piyem-	turning, spiral	2887

Pentland (1979) states that PA postconsonantal *y becomes “y” or “iy” in Shawnee. He cites four examples of each of the two types of reflexes and concludes the section by saying that the “conditioning of the variants is not clear from Voegelin”. The latter of the two, “iy”, was omitted because only one of the two reflexes may be taken for the Model.

Bloomfield reconstructed PA “*pyi:mahamwa” based on Menominee forms. Since the Menominee cognates of the Cree and Ojibway forms, “pi:maham”, etc. and “pi:minakkawa:n”, etc. respectively, contain a long “i”, as in “pi:maham”, Bloomfield reconstructed PA *yi:. This is because if the PA form contained a long “i”, the Menominee reflex would be long “e”. The Shawnee forms with “piyem-” suggest either that the PA form may have been “*piyemahamwa”, or that the Shawnee form underwent an independent development.

Since there are no correspondences to this variation in Shawnee, it is impossible, with present knowledge to determine what has taken place.

2.2.5 Other Rules

Epenthesis:

V > hV / #_

*ameθkwa	hameʔkwa	beaver	0261
*o:te:weni	hoteewe, hoteewena	town, -s	2471
*wetempi	hotepi (nitepi)	his brain (my brain)	3886

Apocope:

V: > V / _#

*kakye:pehše:wa	kakeepeʔše	he is deaf in one ear	0901
*mi:si:wa	miiθi	defecate	1961

In the two cases cited above, the suffix *-wa of AI verbs is lost in Shawnee. This leaves the word final position with a long vowel. This long vowel subsequently becomes short. For a further discussion on suffixes and morphology, see section 2.4.

Syncope:

V > ø / _šC

*eškwe:wi	škote	fire	0682
*maškye:kwi	mškeekowali	lakes	1618
*neški:nšekwi	nʔškiišekwi, nʔškiišeko	my eye, -s	2332

Note that this rule does not affect word initial *we-, (*we- > ho- / #_):

*weškikenwi	hoškinwi	it is brand new	3839
*weškikiwi	hoškiwi	he is brand new	3840

V > ø / [+cons]_?C

*keʔčikamyi	kčikami	ocean	1063
*neʔθwi	nʔθwi	three	2319

V > ø / _hV

*naha:nka	nhaakana	son-in-law	2098
*nahi-	nh-	expertly, curiously	2110
*nekamohe:wa	ninekamha	I sing for him while he dances	2248

Shortening of Long Vowels:

V: > V / _{?C, šC}

*le:hle:weni	le?θeewe	soul, breath	1428
*no:hθa	no?θa	my father	2414
*wi:ʔəniwa	wiʔəni	he is eating	4027

V: > V / _hV

*nana:hi-	nanahi:	preparing (for departure)	2147
*wete:himini	hotehimi	strawberry	3875

Other related phenomena from Pentland (1979)

Related Phenomenon 1

[+cons, +voice] > [-voice] / #_ {h, šC}

*naha:nkana ([ŋ])	nhaakana	son-in-law (Pentland 1979: 164)
-------------------	----------	---------------------------------

Related Phenomenon 2

? > ø / [+cons, -voice]_C

*netahšama:wa	nitsama	I feed him (Pentland 1979: 164)
---------------	---------	---------------------------------

Related Phenomenon 3

? > ø / _CC

*neči:ʔšiha:wa	ničišha	I scare him (Pentland 1979: 164)
----------------	---------	----------------------------------

Related Phenomenon 4

ø > [ʔ] / [+cons, +voice]_ {hV, šC}

*neleškima:wa	nił[ʔ]škima	I scold him (Pentland 1979: 164)
*nepi:nθaha:ni	nipiin[ʔ]ha	I stuck it in (Pentland 1979: 141)

2.2.6 List of all the rules for PHONO

The following is a complete list of all the rules from the preceding sections in which a given

PA segment undergoes a regular sound change.

1. V: > V / _# (apocope)
2. V > hV / #_ (epenthesis)
3. *čk > ʔk
4. *čp > ʔp
5. *ʔl > ʔθ
6. *ʔs > ʔθ
7. *hč > ʔč
8. *hk > ʔk
9. *hl > ʔθ
10. *hp > ʔp
11. *hš > ʔš
12. *hs > ʔθ
13. *hθ > ʔθ
14. *ht > ʔt
15. *i: > i / _y
16. *ke > ki / #_
17. *mp > p
18. *nč > č
19. *nk > k
20. *nl > n
21. *nš > š
22. *ns > θ
23. *nθ > n
24. *nt > t
25. * [+cons, +voice] > [-voice] / #_ {h, šC} (Relatedphenom 1)
26. *ʔ > ø / [+cons, -voice]_C (Relatedphenom 2)
27. *ʔ > ø / _CC (Relatedphenom 3)
28. *ø > [ʔ] / [+cons, +voice]_ {hV, šC} (Relatedphenom 4)
29. *V: > V / _{ʔC, šC} (Shortening1)
30. *V: > V / _hV (Shortening_hV)
31. *sk > šk
32. *V > ø / _šC (Syncope1)
33. *V > ø / [+cons -voice]_ʔC (Syncope2)
34. *V > ø / _hV (Syncope_hV)
35. *s > θ

36.	*θk	>	ʔk
37.	*θp	>	ʔp
38.	*twi	>	ti
39.	*θ	>	l
40.	*wa	>	o / #_
41.	*we	>	o / {C̄, #}_
42.	*xk	>	ʔk
43.	*xp	>	ʔp
44.	*ya:	>	a: / {č, š, w}_
45.	*ye:	>	e: / {č, š}_
46.	*ye	>	i / {-syllabic}_
47.	*yi:	>	i: / {-syllabic}_
48.	*yi	>	i / {-syllabic}_

2.2.7 Rule Order

The following rule order is necessary to prevent the output of one rule interfering with another rule:

*θ	>	l
*s	>	θ

PA *θ changes to Shawnee “l” and PA *s changes to Shawnee “θ”. But the output of the second rule cannot serve as the input of the first rule. In other words, PA *s does not first change to Shawnee “θ” and then undergo a second sound change. Only original PA *θ changes to Shawnee “l”. Since PA *s and *θ are part of many clusters, all other rules must precede these.

In addition to the above rule order, the following rule order is necessary so that the output of one rule may serve as the input of another rule:

*we > o
 *V > hV / # _ (epenthesis)

When PA *we occurs initially, the first rule results in a vowel in the initial position. “h” is epenthesized to all initial vowels, including initial “o” which is a reflex of PA *we. Therefore, the rule for epenthesis must follow the first rule. The complete rule order file has been listed in the next chapter in section 3.4.1.

2.3 Reflexes

This list of PA words and their reflexes in Shawnee is organized as follows: the first line consists of the PA word, its grammatical category, its English gloss and its number in the MD; the second line consists of the Shawnee reflex and its English gloss.

*a:hpawe:wa	AI dream	0015
ha?powe	he dreams	
*ahši:mini	NI pawpaw	0162
ha?šiimi, ha?šiimiiwali	papaw, -s	
*ameθkwa	NA beaver	0261
hame?kwa	beaver	
*ameθkw(V:ns)a	NA small beaver	(Pentland 1979: 143)
ame?kooθa	small beaver	
*amwe:wa	TA eat	0263
hamw-	eating	
*ani:py-	N elm	0268
haniipi, haniipimiši	elm tree	
*anyikwa	NA squirrel	0276
hanikwa	squirrel	
*apwe:wa	AI roast	0301
hapweewa	roasting, barbecuing	
*asa:ki:wa	NA Sauk	0340
hoθaaki	Sauk and Fox	
*ata:wa:wa	NA Odawa	0383
hotaawa	Ottawa	

*axkehkwa	NA kettle	0475
ha7ko7kwa	bucket, kettle, vessel	
*axpapiwa	AI sit on	0508
ha7papi	sitting on something	
*či:hči:peška:wa	AI he trembles	(Pentland 1979: 136)
či7čiipeška	he trembles	
*čya:ki-	PV completely	0593
čaak-, čaak(y)-	group numbering all	
*e:mehkwa:n-	N spoon	0602
heem7kwa, -aanaki	spoon, -s	
*eškwete:wi	NI fire	0682
škote	fire	
*ešpemenki	XP above	0697
špemeki	up, high up (particle)	
*eθpwekwatwi	II taste	0823
7pokwanwi	it tastes in some way	
*ka:wiya	NA quill	0872
kaawi, kaawiiki	briar, -s	
*ka:winšyi	NI bramble, briar, bur	0875
kaawiši	briar plant	
*kakye:pehše:wa	AI he is deaf	0901
kakeepe7še	he is deaf in one ear	
*kečkyatwi	II it is old	(Pentland 1979: 138)
kkatwi	it is old	
*ke7čikamyi	NI sea	1063
kčikami	ocean	
*kešyete:wi	II it is hot	1109
kišite	it is hot	
*ketakesiwa	AI he is spotted, mottled	1116
kitakeθi	he is spotted	
*ki:šesamwa	TI cook	1222
nikiišeθa	I cooked	
*ki:šwawihšinwa	AI warm	1286
kiišo7šinwa	he lies warmly covered	
*ko:na	NA snow	1330
koona	snow	
*kwečpenwa	TI he dreads something	(Pentland 1979: 134)
kwpenwa	he dreads something	
*le:hle:weni	NI breath	1428
le7θeewe	soul, breath	
*le:kawi	NI sand	1430

leekawi	sand	
*lepwa:hka:wa	AI clever	1437
lepwa?ka	he's sober, conscious	
*ma:nθehsi	NI flint	1496
maane?θi	knife	
*maškye:kwi	NI swamp	1618
mškeekowali	lakes	
*meskw(y)e:wi	II it is red	(Pentland 1979: 138)
mškwaawi	it is red	
*meta:hθwi	XP ten	1891
meta?θwi	ten	
*mi:ciθe:wa	TA defecate	1936
nimiičila	I defecate on an. object	
*mi:ciwa	TI eat	1937
miiči	eating (with inan. goal only)	
*mi:si:wa	AI defecate	1961
miiθi	defecate	
*naha:nka	NA son-in-law	2098
nhaakana	son-in-law	
*nahi-	PV well	2110
n?h-	expertly, curiously	
*name:ʔsa	NA fish	2142
name?θa, name?θaki	fish, many fish	
*nana:hi-	PV proper	2147
nanahi:	preparing (for departure)	
*neči:ʔšiha:wa	TA I scare him	(Pentland 1979: 164)
ničišha	I scare him	
*nehta:waka:yi	NI my ear	2241
n?ta:waka	my ear	
*nekamohe:wa	TA he makes him sing	2248
ninekam?ha	I sing for him while he dances	
*nekamowa	AI sing	2249
nekamo	singing	
*nekwetwi	XP one	2258
nekoti, nekotw-	one, alone	
*neleškima:wa	TA I scold him	(Pentland 1979: 164)
nił[ʔ]škima	I scold him	
*nemihsa	NA my elder sister	2266
nimi?θa	my older sister	
*neneθki	NI my arm, hand	2272
(-)neʔki, -n?ke (hone?ki)	arm, wrist (his arm)	

*nepačkyama:wa nipkama	TA I bite off a piece of him (Pentland 1979: 138) I bite off a piece of him	
*nepe:wa nepeewa	AI he sleeps he is sleeping	2282
*nepi:nθaha:ni nipiinha	TI I stuck it in I stuck it in	(Pentland 1979: 141)
*nepwa nepwa, nepooya	AI die he, it is dead	2287
*nepyi nepi	NI water water	2292
*neʔθwi nʔθwi	XP three three	2319
*nesiči niθiči, niθitali	NI my foot my foot, feet	2327
*nešihsa nišiʔθa	NA my cross-uncle my uncle	2328
*neški:nšekwi nʔškiššekwi, nʔškiššeko	NI my eye my eye, -s	2332
*ahšame:wa nitsama	TA I feed him I feed him	0159
*neθenčyi nileči, nilečali	NI my hand my finger, -s	2347
*neθkenθeki n(?)kineki	NI my forehead my forehead	(Pentland 1979: 141)
*nexka:či nkaači, (-)ʔkaat-	NI my leg my leg, leg	2354
*nexka:tali nkaatali	NI my legs my legs	2355
*ni:piči niipiči, niipitali	NI my tooth my tooth, teeth	2399
*no:hθa noʔθa, noʔθaki	NA my father my father-s	2414
*no:nle:wa ninoonepe, ninoonʔha	TA suckle we are suckling, I nurse him	2419
*no:nšye:- no:še	PV have offspring, young she has a child	2420
*nyi:šwi nišwi	XP two two, pair, double, together	2462
*o:te:weni hoteewe, hoteewena	NI town town, -s	2471
*paka:n-	N nut	2513

pakaani	nut	
*pakama:kani	NI cudgel	2514
pakamaaka(na)	war club(s)	
*pake:wa	AI he pounds, he strikes	2518
pake	he pounds, breaks nuts	
*pakiθe:wa	TA he casts her off	2521
nipakila	I throw him away	
*pasekw:wa	AI he arises from sitting	2542
paθekw-	arise	
*pele:wa	NA partridge	2587
peleewa	chicken, turkey, fowl	
*pemi?le:wa	AI fly	2627
pem?θe	he goes by	
*pemotamwa	TI shoot	2639
nipemota	I shoot at it	
*pemwe:wa	TA shoot	2640
pemw-	shoot	
*penkwi	NI ashes	2654
pekw-, pekow-	dust, ashes, worn out, sore.	
*pepikwani	NI whistle	2657
pepikwana	musical instruments	
*pepo:nwi	II winter	2661
pepooni	it is winter	
*pešiwa	NA lynx	2692
pešiwa	wildcat	
*pi:nte:wi	II inside	2743
piit-	in, in something, inside, hole	
*pi:7te:w-	N foam	2779
pi7teewi	it is foaming	
*po:tawe:wa	AI fire	2818
pootawe-	burning wood	
*po:θkama	NA wild American plum	2823
po7kama	peach	
*si:pi:wi	NI river	3001
θiipi, θiipiiwali	river-s	
*ša:hša:k-	PV bare, naked	3017
ša7šaak- (-aw-, -anaw-)	stripped, naked	
*še:pa:wi	XP morning	3060
šeepaawi	morning	
*šeka:kwa	NA skunk	3061
šekaakwa	skunk	

*šekiwa	AI urinate	3067
šekiwa, šekiiki, nišekipe	he, they, we urinate	
*ši:pi:wa	AI stretch	3095
šiipi	stretch	
*ši:7ši:pa	NA duck	3098
ši7šiipa	duck	
*ši:7ši:p(V:ns)a	NA small duck	(Pentland 1979: 143)
ši7šiipeeθa	small duck	
*takwa:kiwi	II autumn	3203
takwaaki	it is autumn	
*te:pwe:wa	AI true	3302
teepwe	he is true, tells the truth	
*tepexki	XP night	3348
tep(e?)ki	night	
*θe7θe:ma:wa	NA tobacco	3394
la7θeema	tobacco	
*wa:pame:wa	TA see	3423
niwaapama	I look at him	
*wa:pančike:wa	AI look	3427
waapačike	he looks on	
*wa:panke	XP tomorrow	3430
waapake	tomorrow	
*wa:pantamwa	TI see	3432
niwaapata	I look at it	
*wa:panwi	II dawn	3437
waapanwi	it is morning	
*wa:pesiwa	NA swan	3447
waapeθi	crane [species not determined]	
*wa:wi	NI egg	3508
howaawi, howaawali	egg, -s	
*walakačkwali	NI his palate	(Pentland 1979: 138)
holaka7kooli	his palate	
*walahakaya	NA scale of fish or reptile	3530
hol7haka	rind, scale	
*wanehše:wa	AI he hears wrong	(Pentland 1979: 160)
wane7še	he hears wrong	
*wanihta:wa	PT lose	3550
niwan?to	I missed it	
*wasye?θanwi	NI nest	3558
wθi?θana	nest	
*wečipwe:wa	NA Ojibwa	3643

hočipwe	Ojibwa	
*wečye:pihki	NI root	3652
hočeepi7ki	medicine	
*wečye:wa	NA fly	3655
hoče, hočeeki	fly, flies	
*wekima:wa	NA chief	3665
hokima, nookimaawi	chief, I am chief	
*wempi-	PV upwards	3719
hop-	up	
*wenči-	PV whence	3726
hoči-	from, away from	
*wenči-eskatwi	II it (water) dries up	(Pentland 1979: 138)
hočiškatwi	it (water) dries up	
*wesa:wya:wi	II yellow	3806
hoθaawa	yellow	
*weškikenwi	II new	3839
hoškinwi	it is brand new	
*weškikiwi	AI young	3840
hoškiwi, nooškiye	he is brand new, we are new	
*weta:hpantamwa	TI suck	3851
noota7pata	I sucked it	
*wete:himini	NI strawberry	3875
hotehimi	strawberry	
*wetempi	NI his brain	3886
hotepi (nitepi)	his brain (my brain)	
*weto:ni	NI his mouth	3888
hotooni, nitooni	his mouth, my mouth	
*weθkani	NI bone	3898
ho7kani, ho7kanali	bone, his bones	
*wexpenya	NA potato	3912
ho7penya, ho7ponyeeki	prewhite sweet potato, -es	
*wi:hpe:me:wa	TA sleep with	3935
niwi7peema	I sleep with him	
*wi:hpe:ntwiwaki	AI sleep with	3937
niwi7peetipe	we sleep together	
*wi:kiwa:mpi	NI house	3945
wiikiwaapali	houses	
*wi:kopyi	NI basswood	3948
wiikopi, wiikopiwali	smooth bark, -s	
*wi:ni	NI marrow	3980
wiini	kernel, marrow	

*wi:nkanwi	II taste good	3989
wiikanwi	it tastes good	
*wi:nle:wa	TA name	3997
niwiina	I named him	
*wi:ntamawike:wa	AI name	4004
niwiitamaake	I was notifying others	
*wi:ntamwa	TI name	4005
niwiita	I named it	
*wi:ʔsakanwi	II bitter	4012
wiʔʔakanwi	it is bitter	
*wi:ʔʔeniwa	AI eat	4027
wiʔʔeni	he is eating	
*wi:ʃi	NI his head	4033
wiʃi	head, his head	
*wi:ʔani	NI his tongue	4049
wiilani	the tongue, his tongue	
*wi:ʔeʔsi	NI hair	4050
wiileʔʔi, wiileʔʔa	head hair-s, his head hair	
*wi:yaka:kamyi	NI dirty water	(Pentland 1979: 161)
wiyakaakami	dirty water	
*wi:yaweysi	NI meat	(Pentland 1979: 161)
wiyawʔʔi	meat	

The fundamental objective of the current study is to identify regular rules of historical sound change and set up a computer model using these rules. In historical change, however, there will always be a certain amount of irregularity. Irregular sound changes, such as metathesis, haplology, dissimilation, etc., are sporadic and do not apply to the entire lexicon and, therefore, they cannot be included in setting up a computer model. Thus, these sound changes will not be discussed in great detail. The irregular sound changes in the list follow; the relevant segment is in bold, and the MD number is cited in brackets:

Haplology: *weš**k**ikenwi > hoškinwi (3839); *weš**k**ikiwi > hoškiwi (3840). Back umlaut: *a:hpawe:wa > haʔpowe (0015); *ax**k**ehkwehsa > haʔkoʔkwa (0477).

The list of reflexes also includes of a few examples in which the prefix and/or the suffix in the PA word does not match that of the Shawnee reflex. Examples: *pakiθe:wa > nipakila (2521); *wi:hpe:ntwiwaki > niwi?peetipe (3937). In 2521, the PA word has no personal prefix, meaning that the word is third person, but the Shawnee prefix is for first person singular number, “ni”. In 3937, the PA word has no personal prefix indicating that the word is third person, and the suffix is for an animate, intransitive verb (AI), third person plural, “waki”. The Shawnee reflex, however, has the prefix for the first person plural inclusive, “ni”, and the suffix for an AI verb, first person plural inclusive, “pe”. This is because either the PA or the Shawnee dictionary does not have the appropriate prefix or suffix in combination with the given word. This is not a problem because the complete paradigms for PA and Shawnee are known. And if only one member of the paradigm is cited in one of these dictionaries for a given word, the remaining members of the paradigm can usually be formed.

2.4 Morphology

There are two points of morphology that must be discussed here. These are the suffixes of the various grammatical categories and the first person pronominal prefix. The first and second person pronominal prefixes in PA are *ne- and *ke-, respectively. These give us Shawnee “ni-” and “ki-”. The problem is that not all initial *ne become “ni”. In other words, a distinction has to be made as to whether an initial *ne is the first person pronominal prefix or the initial segment of the root. On the other hand, the reflex of the second person pronominal prefix, “ki-”, has been regularized and rule 16 applies to all initial *ke. PHONO

is equipped to deal with phonology and not morphology. In other words, it cannot parse the entries and resolve whether an initial *ne is a prefix or part of a root. Thus, such a morphological distinction has to be made manually. In this case, this adjustment can be made either before or after running the PA data in the Shawnee Model.

In the same manner, the suffixes of the various grammatical categories undergo a regular change from PA to Shawnee. Many of the suffixes are identical although they mark different grammatical categories. Thus, some distinction has to be made before the appropriate change can be applied to the given suffix. Therefore, adjustments have to be made for these suffixes manually.

The verbal morphology of PA and Shawnee is tabulated below. These tables serve to assist in adjusting for the morphological changes from PA to Shawnee. They are taken from Goddard (1967b).

2.4.1 Proto-Algonquian Verbal Morphology

TA Objective:

	<i>Direct</i>		<i>Inverse</i>		<i>Inanimate-Subject</i>	
	3	3pl	3	3pl	0	0pl
1	ne--a:wa	-aki	ne--ekwa	-eko:ki	ne--ekweni	-ali
2	ke--a:wa	-aki	ke--ekwa	-eko:ki	ke--ekweni	-ali
3	we--a:wali	-ahi	we--eko:li	-eko:hi	we--ekweni	-ali
1p	ne--a:na:na	-aki	ne--ekwena:na	-aki	ne--ekwena:ni	-ali
12	ke--a:nawa	-aki	ke--ekwenawa	-aki	ke--ekwenawi	-ali
2p	ke--a:wa:wa	-aki	ke--ekowa:wa	-aki	ke--ekowa:wi	-ali
3p	we--a:wa:wali	-ahi	we--ekowa:wali	-ahi	we--ekowa:wi	-ali

TA Absolute:

	<i>Direct</i>	<i>Inverse</i>	<i>Passive</i>	<i>Inanimate-Subject</i>
1	ne--a		ne--eko:Pi	ne--ekwi
2	ke--a		ke--eko:Pi	ke--ekwi
3	--e:wa	--ekwa	--a:wa	--ekwiwa
1p	ne--a:Pena		ne--eko:Pena	ne--ekwiPena
12	ke--a:Pena		ke--eko:Pena	ke--ekwiPena
2p	ke--a:Pewa		ke--eko:Pwa	ke--ekwiPwa
3p	--e:waki	--eko:ki	--a:waki	--ekwiwaki

	<i>TI Objective</i>		<i>TI Absolute</i>	<i>AI</i>	<i>TA You-and-Me</i>	
	3	3pl				
1	ne--a:ni	-ali	ne--a	ne--	2-1	ke--i
2	ke--a:ni	-ali	ke--a	ke--	2p-1	ke--iPwa
3	we--a:ni	-ali	--amwa	--wa	2(p)-1p	ke--iPena
1p	ne--e:na:ni	-ali	ne--a:Pena	ne--Pena		
12	ke--e:nawi	-ali	ke--a:Pena	ke--Pena	1-2	ke--eθe
2p	ke--e:wa:wi	-ali	ke--a:Pwa	ke--Pwa	1-2p	ke--eθePwa
3p	we--e:wa:wi	-ali	--amo:ki	--waki	1p-2(p)	ke--eθePena

2.4.2 Shawnee Verbal Morphology

<i>TA Direct</i>		<i>TA Inverse</i>		<i>AI</i>	<i>ps AI</i>
3	3pl	3	3pl		
1	ni--a -a:ki	ni--ekwa	-eko:ki	ni--	ni--a
2	ki--a -a:ki	ki--ekwa	-eko:ki	ki--	ki--a
3	ho--a:li -ahi	ho--eko:li	-ekohi	-(wa)	--amwa
1p	ni--a:pe	ni--ekona	-ekona:ki	ni--pe	ni--a:pe
12	ki--a:pe	ki--ekona	-ekona:ki	ki--pe	ki--a:pe
2p	ki--a:wa -a:wa:ki	ki--ekowa	-ekowa:ki	ki--pwa	ki--a:pwa
3p	ho--a:wa:li -a:wahi	ho--ekowa:li	-ekowahi	--(wa)ki	--amo:ki

<i>TI</i>		<i>TA Inan.-Subj.</i>		<i>TA Passive</i>	<i>You-and-Me Forms</i>	
3	3pl	0	0pl			
1	ni--a -a:na	ni--eko	-ekona	ni--eko:pi	2-1	ki--i
2	ki--a -a:na	ki--eko	-ekona	ki--eko:pi	2p-1	ki--ipwa
3	ho--a -a:na	ho--eko	-ekona	--o:θo	2(p)-1p	ki--ipe
1p	ni--a:pe	ni--ekope		ni--eko:pe		
12	ki--a:pe	ki--ekope		ki--eko:pe	1-2	ki--ele
2p	ki--a:na:wa	ki--ekona:wa		ki--eko:pwa	1-2p	ki--elepwa
3p	ho--a:na:wa	ho--ekona:wa		--o:θo:ki	1p-2(p)	ki--elepe

A few examples of how these morphological adjustments are applied to the data follow. In the PA table above, the paradigm of the Animate Intransitive AI indicates that a verb in the third person singular gets the suffix “wa”; and that the verb has no personal prefix. The Shawnee table above indicates that the AI verb in the third person singular has the suffix “(wa)”; and that the verb has no personal prefix. Thus, in PA an AI verb in the third person singular has the form: verb-wa; and in Shawnee: ø-verb-ø. Therefore, the suffixes of AI verbs in PA may be marked as follows: *te:pwe:-wa (MD 3302), *kakyē:pehše:-wa (MD 0901). These give us, respectively, Shawnee: teepwe-ø, kakepe?še-ø. Consequently, the AI verbs have to be adjusted by deleting the suffix “wa” from the PA word in order to obtain the

appropriate Shawnee form. Some short verbs do not lose the suffix but most of them do.

Many of the reflexes in the list in this chapter, as indicated in section 2.3, are not in the same person and number as the PA word. For example, the MD consistently cites TI verbs in the Absolute form, third person singular: suffix, “amwa”; no prefix. But the Shawnee dictionary cites TI verbs in the Objective form, in either the third person or first person singular: suffix, “a”; prefix, “ni” or “ho”. The affixes are marked by a dash (-) in the following examples:

*ki:šes-amwa	TI cook	1222
ni-kiišeθ-a	I cooked it	
*pemot-amwa	TI shoot	2639
ni-pemot-a	I shoot at it	

The Shawnee paradigm which corresponds to the PA TI Absolute is the pseudo AI (ps AI): these two paradigms are a perfect match. In other words, the affixes of the citation member of the paradigm, the third person singular, are identical for both PA and Shawnee: no prefix and suffix “-amwa”. This grammatical category, therefore, needs no adjustments. Similarly, the TA verbs are always cited in the MD in the Absolute form: no prefix and suffix “-e:wa”. And Voegelin (1938) cites TA verbs in the Objective form: prefix “ni-” and suffix “-a” (first person singular). Examples:

*mi:čiθ-e:wa	TA defecate	1936
ni-miičil-a	I defecate on an. object	
*pakiθ-e:wa	TA he casts her off	2521
ni-pakil-a	I throw him away	

It follows that the TA verbs ending in “-e:wa” have to be adjusted by deleting this suffix from

the PA word and adding the prefix, “ni-” and the suffix “-a” to obtain the first person singular Shawnee form. Of course, any other member of the TA Direct paradigm can be obtained in this way.

By making such changes, the morphological reflexes for all the grammatical categories can be worked out. However, this requires an excessive amount of morphological work especially for the intransitive verbs, and the nouns which have larger lists of possible suffixes. Therefore, for this study, the PA suffixes were left unaltered except for the regular phonological rules.

It is true that if all the verbs are properly labeled, PHONO might be able to provide the correct ending or suffix in the same manner it applies all the other rules. But PHONO is currently not programmed to handle extensive morphological changes. Thus, to set up a Model which can provide the correct endings would require too many rules. Each of these rules would take PA entries with a given grammatical category, person and number and transform it to the corresponding Shawnee form. Moreover, all the PA entries would have to be marked for grammatical category as well as person and number. Although it is desirable to have features in PHONO which would permit us to write paradigms plus simple morphological rules to provide the necessary endings, this feature is not currently available in PHONO.

Chapter Three:

PHONO - Setting up the Shawnee Model

3.0 Introduction

This chapter deals with the program, PHONO, and the various steps required to set up a model. The model that was set up is called the Shawnee Model. All the files associated with this Model are named "shawnee.*"; the asterisk marks the appropriate file extension required by PHONO. The first section is a general overview of the program. In the next three sections, the three basic files required to set up and run a model in PHONO are described and illustrated by examples from the Shawnee Model. The last section gives a brief description of some of the features of PHONO.

3.1 Introduction to PHONO

The program runs on DOS. However, for most of my work, I have run it on a DOS emulator on Windows 95 and did not face any additional difficulties. Of course, the regular difficulties are that like all other programs, it crashes from time to time. The frequency is slightly higher than popular programs, such as WordPerfect, because it has been written and maintained by a single person. As a result, it has not been debugged as much as the popular programs.

Several models can be set up in the program and these would be available for use on running PHONO. The only criterion is that all the files of a given model have the same name; of

course, each of these files would require different extensions. Only one model may be loaded at a time. Once a model is loaded, all its files are available in the menu for use or for editing. Since all the files are text documents, we can edit them with any word processor as well if the need arises. It stores all output in temporary files. These may be saved as text documents as desired. It is reasonably fast and is able to run a thousand words in less than one minute.

3.2 The Alphabet File

PHONO has the capacity to handle up to 70 different characters selected from among the 255 ASCII characters. However, both PA and Shawnee together require a total of about 25 different characters — i.e. far less than the total inventory of characters available in PHONO. A set of up to 23 binary features can be named by the user; for example: consonant, high, round, etc. Following the Spanish Model which is provided with the program, 18 binary features have been used for the Shawnee Model. Finally for each of the characters, the desired binary values (+/-) can be specified. The feature values appear in the Alphabet file as a single (+/-) or double (#!/=) sign. There is no difference, in principle, between the single signs and the double signs. However, the distinction is fundamental to the program because it is with this key that PHONO sorts the characters and executes functions specified in the rules (see Appendix 3, section 4.1).

Although the phonemic inventory for the Shawnee Model is small, a few special characters which are not available in the ASCII character set are required. This does not pose a major

problem but some adjustments are required. The following are the special character correspondences between the phonemic inventory for the Shawnee Model and the characters from the ASCII character set that may be used to represent them while running PHONO:

Phoneme		Symbol used for PHONO
š	=	\$
č	=	c
ʔ	=	·
θ	=	d

Vowel length is indicated by the colon, ":".

One problem faced initially with respect to the Alphabet file was the orthographic representation of theta, "θ". Although theta is an ASCII character, it cannot be entered from the keyboard. To represent theta, "d" could be used in one of two ways: it could be given the binary feature values of theta or it could be changed to a theta by an orthographic rule. The latter option was used because there is a provision in PHONO to write orthographic rules. Thus theta is entered as "d" and an orthographic rule changes "d" to theta, "θ".

3.2.1 The Alphabet File for the Shawnee Model

	a	c	e	ε	h	i	k	l	m	n	o	p	q	s	\$	t	d	θ	u	w	x	y	ʼ	˘	:
cons	=	+	=	=	+	=	+	+	+	+	=	+	+	+	+	+	+	+	=	-	+	=	=	+	=
syllabic	#	-	#	#	=	#	-	-	-	-	#	-	-	-	-	-	-	-	#	=	-	=	-	-	=
obstr	=	#	=	=	=	=	#	=	=	=	#	#	#	#	#	#	#	#	=	-	#	=	#	+	=
high	=	#	=	=	-	#	#	=	-	=	=	-	=	=	#	-	-	-	#	+	+	#	-	-	=
low	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#	=
back	#	-	=	=	-	=	#	-	-	-	#	-	#	-	-	-	-	-	#	+	#	=	-	-	=
round	=	-	=	=	-	=	-	-	-	-	#	-	-	-	-	-	-	-	#	#	-	=	-	-	=
coronal	-	#	-	-	=	-	=	#	=	#	-	=	=	#	#	#	#	#	-	-	-	-	=	-	=
anterior	-	-	-	-	=	-	-	+	#	#	-	#	=	#	=	#	#	#	-	-	-	-	=	-	=
distrib	+	+	+	+	+	+	+	+	#	+	+	+	+	+	+	+	+	-	+	+	+	+	-	-	=
cont	+	=	+	+	#	+	=	=	=	=	+	=	=	#	#	=	=	#	+	#	#	+	=	#	=
delrel	+	#	+	+	+	+	-	+	+	+	+	-	-	+	+	-	-	+	+	+	+	+	-	+	=
strident	-	#	-	-	-	-	-	-	-	-	-	-	-	#	#	=	=	=	-	-	-	-	-	-	=
voice	+	=	+	+	+	+	+	+	+	+	+	+	=	=	=	=	#	=	+	+	+	+	-	+	=
nasal	=	-	=	=	-	=	=	-	#	#	=	-	-	-	-	-	-	-	=	-	-	-	-	-	=
tense	+	+	#	=	-	#	+	-	-	-	-	#	+	+	+	+	+	+	#	+	+	+	-	-	=
long	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#
stress	=	-	=	=	-	=	=	-	=	=	-	=	-	-	-	-	-	-	=	-	-	-	-	-	=

The above inventory consists of four extra characters (ε, q, u, and ʼ) which are not required for the Shawnee Model. This inventory was set up with these extra characters so that it might serve as a default file for other models in the Algonquian family of languages. Stress

is an important aspect of diachronic study. PHONO has the capacity to handle stress. Although stress rules were not included in the Shawnee Model, this feature has been included in the Model for future expansion and refinement.

3.3 The Rule Makeup: phonological rules

PHONO's rule notation system is based on standard linguistic rule notation expressions such as the following: $A > B / C _ D$. This means that element A changes into element B in the environment following element C and preceding element D; in other words, each occurrence of the sequence CAD (known as the rule's "structural description") is changed into CBD (and the change of A to B is called the "structural change").

PHONO's rule notation portrays the structural description as a set of if-lines, and the structural change as a set of then-lines. The format of a rule consists of four fields:

1. the name line, consisting of the rule's name (with a limit of 20 characters)
2. any number (up to 25) of if-lines, labeled in order by letters (A, B, C...)
3. any number (up to 25) of then-lines, labeled in order by numerals (1, 2, 3...)
4. the end line, consisting of the word "END" plus the rule name repeated.

For example, the rule, $(V > hV / \# _)$, is written in the Rule Makeup file in the Shawnee Model as follows:

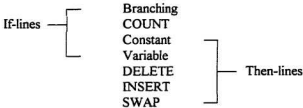
EPENTHETIC_H

```

A: -cons(1) +syllabic(1)
I: INSERT h (1)
END EPENTHETIC_H

```

There are four types of if-lines and five types of then-lines, the Constant and Variable types available for both functions. There are a total of seven types, as shown below:



If the rule has more than one if-line, these must be interrelated in a hierarchy, through the use of Branching-type lines. In contrast, the then-lines are simply applied cumulatively and in order.

If- and then-lines are composed mainly of feature-names and location-expressions. These two building-blocks are used to construct the seven types of lines. With these seven types of lines, virtually all kinds of sound changes may be written.

3.3.1 The Rule Makeup File for the Shawnee Model

The Phonological Rules of Proto-Algonquian reflexes in Shawnee that are listed in the previous chapter have all been written in the Shawnee Model. The Rule Makeup file allows capital letters only and the symbol for underscore, “_”. This is why the name chosen for

each rule is simple and reflective of the change taking place to the given segment. This permits us to keep track of the rules and locate and solve problems without difficulty. However, because of the limit on the choice of characters in naming rules, “7” is written as “G”, as in glottal stop, and “θ” as “TH”, as in theta and so on.

PHONO sorts the Rule Makeup file alphabetically on the name of each rule and this is the order in which the rules are presented below.

Each rule has been written as simply as possible. In two cases, the rules have been split in two, Rule 39, 39a and Rule 41, 41a, because it seems that this is a simpler way to write these rules.

The Rule Makeup file, “shawnee2.mak” follows with the addition of the rule number and the rule, at the beginning of each rule, as they appear in the previous chapter:

1. V: > V / _# (apocope)

APOCOPE

A: B and C

B: +long(-1)

C: +syllabic(-2)

1: DELETE (-1)

END APOCOPE

2. *čk > ?k

CHK_BECOMES_GK

A: +obstr(*) -cont(*) +coronal(*) -anterior(*) +high(*+1) +back(*+1) -cont(*+1)

1: +cont(*) -coronal(*) -high(*) +low(*) -distrib(*) -strident(*) +voice(*) -tense(*)

END CHK_BECOMES_GK

3. *čp > ʔp
 CHP_BECOMES_GP
 A: +obstr(*) -cont(*) +coronal(*) -anterior(*) +obstr(*+1) +anterior(*+1) -coronal(*+1)
 I: -high(*) +low(*) -coronal(*) -distrib(*) +cont(*) -strident(*) +voice(*) -tense(*)
 END CHP_BECOMES_GP

d > θ (Orthographic rule, not in previous chapter and so not numbered here)
 D_TO_THETA
 A: +obstr(*) -cont(*) +voice(*)
 I: -distrib(*) +cont(*) +delrel(*) -voice(*) +tense(*)
 END D_TO_THETA

4. V > hV / #_ (epenthesis)
 EPENTHETIC_H
 A: -cons(1) +syllabic(1)
 I: INSERT h (1)
 END EPENTHETIC_H

5. *ʔl > ʔθ
 GL_BECOMES_GTH
 A: +voice(*) -cont(*) -nasal(*) -obstr(*) +cons(*-1) +low(*-1)
 I: -voice(*) +cont(*) +tense(*) +obstr(*) -distrib(*)
 END GL_BECOMES_GTH

6. *ʔs > ʔθ
 GS_BECOMES_GTH
 A: +cont(*) +anterior(*) +strident(*) +cons(*-1) +low(*-1)
 I: -strident(*) -distrib(*)
 END GS_BECOMES_GTH

7. *hč > ʔč
 HCH_BECOMES_GCH
 A: +cons(*) -obstr(*) +cont(*) -low(*) +obstr(*+1) -cont(*+1) +coronal(*+1) -anterior(*+1)
 I: +obstr(*) +low(*) -distrib(*) +voice(*)
 END HCH_BECOMES_GCH

8. *hk > ʔk
 HK_BECOMES_GK
 A: +high(*) +back(*) -cont(*) +cons(*-1) +cont(*-1) -obstr(*-1) -low(*-1)
 I: +obstr(*-1) +low(*-1) -distrib(*-1) +voice(*-1)

END HK_BECOMES_GK

9. *hl > ʔθ

HL_BECOMES_GTH

A: +voice(*) -cont(*) -nasal(*) -obstr(*) +cont(*-1) -obstr(*-1) +cons(*-1)

I: -voice(*) +cont(*) +tense(*) +obstr(*) -distrib(*) +obstr(*-1) +low(*-1) -distrib(*-1) +voice(*-1)

END HL_BECOMES_GTH

10. *hp > ʔp

HP_BECOMES_GP

A: B and C

B: +obstr(*) +anterior(*) -coronal(*)

C: +cons(*-1) -obstr(*-1) +cont(*-1)

I: +obstr(*-1) +low(*-1) -distrib(*-1) +voice(*-1)

END HP_BECOMES_GP

11. *hš > ʔʃ

HSH_BECOMES_GSH

A: +obstr(*) +cont(*) +coronal(*) -anterior(*) +cons(*-1) +cont(*-1) -obstr(*-1) -low(*-1)

I: +obstr(*-1) +low(*-1) -distrib(*-1) +voice(*-1)

END HSH_BECOMES_GSH

12. *hs > ʔθ

HS_BECOMES_GTH

A: +cont(*) +strident(*) +anterior(*) +cont(*-1) -obstr(*-1) +cons(*-1)

I: -voice(*) -strident(*) -distrib(*) +low(*-1) +obstr(*-1) -distrib(*-1) +voice(*-1)

END HS_BECOMES_GTH

13. *hθ > ʔθ

HTH_BECOMES_GTH

A: +coronal(*) -distrib(*) +cons(*-1) -obstr(*-1) +cont(*-1)

I: +obstr(*-1) +low(*-1) -distrib(*-1) +voice(*-1)

END HTH_BECOMES_GTH

14. *ht > ʔt

HT_BECOMES_GT

A: +obstr(*) +coronal(*) -delrel(*) -voice(*) +cont(*-1) +cons(*-1) -obstr(*-1) -low(*-1)

I: +cont(*-1) +cons(*-1) +obstr(*-1) +low(*-1) -distrib(*-1) +voice(*-1)

END HT_BECOMES_GT

15. *i: > i / _y

II_BECOMES_I

A: B and C

B: -cons(*) +syllabic(*) +high(*) -back(*) -low(*) +long(*+1)

C: -cons(*+2) -syllabic(*+2) +high(*+2) -back(*+2) -round(*+2)

I: DELETE (*+1)

END II_BECOMES_I

16. *ke > ki / #_C

KEI_BECOMES_KI

A: B and C

B: +high(1) +back(1) -cont(1)

C: -cons(2) +syllabic(2) -high(2) -low(2) -back(2) +tense(2) -syllabic(3) -long(3)

I: +high(2)

END KEI_BECOMES_KI

17. *mp > p

MP_BECOMES_P

A: +obstr(*) +anterior(*) -coronal(*) +nasal(*-1) -coronal(*-1)

I: DELETE (*-1)

END MP_BECOMES_P

18. *nč > č

NCH_BECOMES_CH

A: +obstr(*) -cont(*) +coronal(*) -anterior(*) +nasal(*-1) +coronal(*-1)

I: DELETE (*-1)

END NCH_BECOMES_CH

19. *nk > k

NK_BECOMES_K

A: +back(*) -cont(*) +high(*) +nasal(*-1) +coronal(*-1)

I: DELETE (*-1)

END NK_BECOMES_K

20. *nl > n

NL_BECOMES_N

A: -obstr(*) -cont(*) +voice(*) -nasal(*) +nasal(*-1) +coronal(*-1)

I: DELETE (*)

END NL_BECOMES_N

21. *nš > š

NSH_BECOMES_SH
 A: +obstr(*) +cont(*) +coronal(*) -anterior(*) +nasal(*-1) +coronal(*-1)
 1: DELETE (*-1)
 END NSH_BECOMES_SH

22. *ns > θ
 NS_BECOMES_TH
 A: +cont(*) +anterior(*) +strident(*) +nasal(*-1) +coronal(*-1)
 1: -voice(*) -strident(*) -distrib(*)
 2: DELETE (*-1)
 END NS_BECOMES_TH

23. *nθ > n
 NTH_BECOMES_N
 A: +coronal(*) -distrib(*) +nasal(*-1) +coronal(*-1)
 1: DELETE (*)
 END NTH_BECOMES_N

24. *nt > t
 NT_BECOMES_T
 A: +obstr(*) +coronal(*) -delrel(*) -voice(*) +nasal(*-1) +coronal(*-1)
 1: DELETE (*-1)
 END NT_BECOMES_T

25. *[+cons, +voice] > [-voice] / #_{h, šC} (Relatedphenom 1)
 RELATEDPHENOM1
 A: B and E
 B: C or D
 C: +cons(2) +low(2) +cons(3)
 D: +obstr(2) +cont(2) +coronal(2) -anterior(2) +cons(3)
 E: -syllabic(1) +voice(1)
 1: -voice(1)
 END RELATEDPHENOM1

26. *ʔ > ø / [+cons, -voice]_C (Relatedphenom 2)
 RELATEDPHENOM2
 A: B and C
 B: +cons(*) +low(*)
 C: -syllabic(*-1) -voice(*-1) +cons(*+1)
 1: DELETE (*)
 END RELATEDPHENOM2

27. *ʔ > ø / _CC (Relatedphenom 3)

RELATEDPHENOM3

A: B and C

B: +cons(*) +low(*)

C: +cons(*+1) +cons(*+2)

l: DELETE (*)

END RELATEDPHENOM3

28. *ø > [ʔ] / [+cons, +voice]_ {hV, ʃC} (Relatedphenom 4)

RELATEDPHENOM4

A: B and E

B: C or D

C: +cons(*+1) -obstr(*+1) +cont(*+1) +syllabic(*+2)

D: +obstr(*+1) +cont(*+1) +coronal(*+1) -anterior(*+1) +cons(*+2)

E: -syllabic(*) +voice(*)

l: INSERT ` (*+1)

END RELATEDPHENOM4

29. *V: > V / _{ʔC, ʃC} (Shortening1)

SHORTENING1

A: B and E

B: C or D

C: +cons(*) +low(*) +cons(*+1)

D: +obstr(*) +cont(*) +coronal(*) -anterior(*) +cons(*+1)

E: +long(*-1)

l: DELETE (*-1)

END SHORTENING1

30. *V: > V / _hV (Shortening_hV)

SHORTENING_HV

A: +cons(*) -obstr(*) +cont(*) -cons(*+1) +long(*-1)

l: DELETE (*-1)

END SHORTENING_HV

31. *sk > ʃk

SK_BECOMES_SHK

A: +high(*) +back(*) -cont(*) +cont(*-1) +anterior(*-1) +strident(*-1)

l: +high(*-1) -anterior(*-1)

END SK_BECOMES_SHK

32. *V > ø / _ʃC (Syncopel1)

SYNCOPE1

A: B and C

B: +obstr(*) +cont(*) +coronal(*) -anterior(*) +cons(*+1)

C: +syllabic(*-1)

1: DELETE (*-1)

END SYNCOPE1

33. *V > ø / [+cons -voice]_?C (Syncope2)

SYNCOPE2

A: B and C

B: +cons(*) +low(*) +cons(*+1)

C: +syllabic(*-1) +cons(*-2) -voice(*-2)

1: DELETE (*-1)

END SYNCOPE2

34. *V > ø / _hV (Syncope_hV)

SYNCOPE_HV

A: +cons(*) -obstr(*) +cont(*) -cons(*+1) +syllabic(*-1)

1: DELETE (*-1)

END SYNCOPE_HV

35. *s > θ

S_BECOMES_TH

A: +cont(*) +anterior(*) +strident(*)

1: -voice(*) -strident(*) -distrib(*)

END S_BECOMES_TH

36. *θk > ʔk

THK_BECOMES_GK

A: +high(*) +back(*) -cont(*) +coronal(*-1) -distrib(*-1)

1: +low(*-1) -coronal(*-1) -anterior(*-1) +voice(*-1) -tense(*-1)

END THK_BECOMES_GK

37. *θp > ʔp

THP_BECOMES_GP

A: B and C

B: +obstr(*) +anterior(*) -coronal(*)

C: +coronal(*-1) -distrib(*-1)

1: +low(*-1) -coronal(*-1) -anterior(*-1) +voice(*-1) -tense(*-1)

END THP_BECOMES_GP

38. *twi > ti
 TWI_BECOMES_TI
 A: B and C
 B: D and E
 C: +syllabic(*+1) +high(*+1) -back(*+1)
 D: -cons(*) -syllabic(*) +high(*) +back(*) +round(*)
 E: +obstr(*-1) +coronal(*-1) -delrel(*-1) -voice(*-1)
 I: DELETE (*)
 END TWI_BECOMES_TI

39. *θ > l / #_
 VTHI_BECOMES_L
 A: B and C
 B: +obstr(1) -distrib(1) +cont(1) +delrel(1) -voice(1) +tense(1)
 C: -cons(2) +syllabic(2) -obstr(2)
 I: -obstr(1) +distrib(1) -cont(1) +voice(1) -tense(1)
 END VTHI_BECOMES_L

39a. *θ > l / V_V
 VTHV_BECOMES_VLV
 A: B and G
 B: C and D
 C: E or F
 D: +obstr(*) -distrib(*) +cont(*) +delrel(*) -voice(*) +tense(*)
 E: +long(*-1)
 F: -cons(*-1) +syllabic(*-1) -obstr(*-1)
 G: -cons(*+1) -obstr(*+1)
 I: -obstr(*) +distrib(*) -cont(*) +voice(*) -tense(*)
 END VTHV_BECOMES_VLV

40. *wa > o / #_
 WA1_BECOMES_O
 A: B and E
 B: C and D
 C: -cons(1) -syllabic(1) +high(1) +back(1) +round(1)
 D: -cons(2) +syllabic(2) -high(2) +low(2) +back(2)
 E: -syllabic(3) -long(3)
 I: -low(2) +round(2)
 2: DELETE (1)
 END WA1_BECOMES_O

41. *we > o / #_

WE1BECOMESO

A: B and E

B: C and D

C: -cons(1) -syllabic(1) +high(1) +back(1) +round(1)

D: -cons(2) +syllabic(2) -high(2) -low(2) -back(2) +tense(2)

E: -syllabic(3) -long(3)

1: +back(2) +round(2)

2: DELETE (1)

END WE1BECOMESO

41a. *we > o / C_

WEC_BECOMES_OC

A: B and E

B: C and D

C: F and G

D: +cons(*-1) -syllabic(*-1)

E: -cons(*) -syllabic(*) +high(*) +back(*) +round(*)

F: -cons(*+1) +syllabic(*+1) -high(*+1) -low(*+1) -back(*+1) +tense(*+1)

G: -syllabic(*+2) -long(*+2)

1: +back(*+1) +round(*+1)

2: DELETE (*)

END WEC_BECOMES_OC

42. *xk > ʔk

XK_BECOMES_GK

A: -cont(*) +back(*) +high(*) +obstr(*-1) +cont(*-1) -low(*-1) +back(*-1)

1: -distrib(*-1) +voice(*-1) -tense(*-1) -high(*-1) +low(*-1) -back(*-1)

END XK_BECOMES_GK

43. *xp > ʔp

XP_BECOMES_GP

A: +obstr(*) +anterior(*) -coronal(*) +obstr(*-1) +back(*-1) +cont(*-1)

1: -distrib(*-1) +voice(*-1) -tense(*-1) -high(*-1) +low(*-1) -back(*-1)

END XP_BECOMES_GP

44. *ya: > a: / {č, š, w}_

YAA_BECOMES_AA

A: B and I

B: C and F

C: D and E

D: G or H
 E: -cons(*) -syllabic(*) +high(*) -back(*) -round(*)
 F: -cons(*+1) +syllabic(*+1) -high(*+1) -low(*+1) +back(*+1)
 G: +high(*-1) +strident(*-1)
 H: -cons(*-1) -syllabic(*-1) +high(*-1) +back(*-1) +round(*-1)
 I: +long(*+2)
 1: DELETE (*)
 END YAA_BECOMES_AA

45. *ye: > e: / {č, š}_
 YEE_BECOMES_EE
 A: B and G
 B: C and D
 C: E and F
 D: -syllabic(*-1) -long(*-1)
 E: -cons(*) -syllabic(*) +high(*) -back(*) -round(*)
 F: -cons(*+1) +syllabic(*+1) -high(*+1) -low(*+1) -back(*+1) +tense(*+1)
 G: +high(*-1) +strident(*-1) +long(*+2)
 1: DELETE (*)
 END YEE_BECOMES_EE

46. *ye > i / {-syllabic}_
 YE_BECOMES_I
 A: B and C
 B: D and E
 C: -syllabic(*-1) -long(*-1)
 D: -cons(*) -syllabic(*) +high(*) -back(*) -round(*)
 E: -cons(*+1) +syllabic(*+1) -high(*+1) -low(*+1) -back(*+1) +tense(*+1) -syllabic(*+2) -long(*+2)
 1: +high(*+1)
 2: DELETE (*)
 END YE_BECOMES_I

47. *yi: > i: / {-syllabic}_
 YII_BECOMES_II
 A: B and E
 B: C and D
 C: F and G
 D: -syllabic(*-1) -long(*-1)
 E: -cons(*) -syllabic(*) +high(*) -back(*) -round(*)
 F: -cons(*+1) +syllabic(*+1) +high(*+1) -back(*+1) -low(*+1)

```

G: +long(*+2)
I: DELETE (*)
END YI_BECOMES_II

48. *yi > i / {-syllabic}_
YI_BECOMES_I
A: B and C
B: D and E
C: -syllabic(*-1) -long(*-1)
D: -cons(*) -syllabic(*) +high(*) -back(*) -round(*)
E: -cons(*+1) +syllabic(*+1) +high(*+1) -back(*+1) -low(*+1)
I: DELETE (*)
END YI_BECOMES_I

```

3.4 The Rule Order file

The Rule Order file displays two lists: the chronological Order list and the alphabetized Supply list. The Supply list consists of all the rules in the Rule Makeup file. Rules are chosen from the Supply list and installed in the Order list. The Order list determines how the model will be operated: which rules will be applied, and in what order. The rules from Supply may be used in Order once, more than once, or not at all. The Order list is structured in three labeled sections, corresponding to the three major types of rules that PHONO recognizes: (1) Old Orthographic, (2) Phonological, and (3) New Orthographic. The Phonological rules belong to the sound-change model proper, while both kinds of Orthographic rules merely serve to interpret character input.

Old Orthographic rules serve to adjust the feature values of ancestor word input, if such adjustment is required by a given model. For example, Latin orthography needs to be

adjusted before sound change rules can be applied to the ancestor language because there is a discrepancy between Latin orthography and the spoken language. The Phonological rules form the significant body of the sound-change model. Such rules are normally considered transient (acting no more than once in a derivation). They can also be marked persistent (repeating their changes throughout the derivation whenever their conditions occur) according to the needs of a given model. New Orthographic rules have an interpretive function, with regard to the spelling of the known descendant words in the pairs tested and can be applied depending on the needs of the given model. For example, the Spanish Model would have, as an output, the phonemic representation of the given words as they are spoken. There is a discrepancy between Spanish orthography and spoken Spanish. Thus, the output may be modified with New Orthographic rules.

3.4.1 The Rule Order file for the Shawnee Model

The orthography for Proto-Algonquian is phonemic, and so, there is no need for Old Orthographic rules. Since the output is phonemic, there is no need to modify it because the Shawnee data is also phonemic, hence, there is no need for New Orthographic rules. Instead, the provision for orthographic rules have been utilized to overcome the shortage of characters of standard computer keyboards, as discussed in section 3.2 above. The following is the Rule Order file for the Shawnee Model:

ORTHOGRAPHIC
 d_to_theta
 PHONOLOGICAL
 chk_becomes_gk
 chp_becomes_gp
 gl_becomes_gth
 gs_becomes_gth
 hch_becomes_gch
 hk_becomes_gk
 hl_becomes_gth
 hp_becomes_gp
 hsh_becomes_gsh
 hs_becomes_gth
 hth_becomes_gth
 ht_becomes_gt
 kel_becomes_ki
 mp_becomes_p
 nch_becomes_ch
 nk_becomes_k

nl_becomes_n
 nsh_becomes_sh
 nth_becomes_n
 nt_becomes_t
 sk_becomes_shk
 thk_becomes_gk
 thp_becomes_gp
 xk_becomes_gk
 xp_becomes_gp
 vthv_becomes_vlv
 vthl_becomes_l
 ns_becomes_th
 s_becomes_th
 ii_becomes_i
 twi_becomes_ti
 wal_becomes_o
 wec_becomes_oc
 welbecomeso
 yii_becomes_ii

yee_becomes_ee
 yaa_becomes_aa
 yi_becomes_i
 ye_becomes_i
 syncope1
 syncope2
 syncope_hv
 shortening1
 shortening_hv
 relatedphenom1
 relatedphenom2
 relatedphenom3
 relatedphenom4
 epenthetic_h
 apocope
 ORTHOGRAPHIC
 d_to_theta

3.5 Running PHONO

3.5.1 Interactive mode

In this mode, single proto-forms may be entered and the rules applied. The output shows the result. There are options to show the rules and the order in which they were applied. This mode is used extensively for working out and writing the rules. After each rule is written in the Rule Makeup file, it has to be tested to verify that it does what it is expected to do. Various proto-words requiring the application of the given rule to obtain the reflex are run in the interactive mode. The rule is modified whenever a run indicates that the rule has changed a segment that it was not supposed to change. The rule trace option is particularly helpful because it indicates the phonological features that are incorrectly

described in the Rule Makeup.

3.5.2 Batch mode

In this mode, a group of proto-forms are listed along with the corresponding daughter forms, in a Word Pair file, such as "shawnee2.par". When the Word Pair file is run, the output shows the result of the run. The resulting output may consist of good output and bad output. This actually reflects whether there is a match between the daughter forms presented and the result of applying the rules on the proto-form in the given order. Thus, by the process of trial and error, all the rules written in PHONO may be checked along with the rule ordering. This is facilitated by options which show all the rules that were applied to a particular word and the order in which they were applied.

The problem with the batch mode, however, is that the output is stored in a temporary file and is deleted or written-over every time the Word Pair file is run. To get around this problem, the temporary files may be appropriately named and saved in either DOS or any wordprocessor after exiting PHONO.

The Batch Mode was used extensively to ensure that all the rules were properly written in the Shawnee Model and that they applied to the data in the correct order. The entire list of reflexes in section 2.3 was used for batch runs every time a new rule was added to the Model or the rule order was changed. The problem of irregular rules or morphological rules which

have not been included in the Model were taken into account by making the necessary adjustments manually to the Word Pair file.

3.5.3 Tracing the rules

The program provides an option to record the rules of derivation of a descendant form from the proto-form. A chronological list of the rules that act on a given proto-word is recorded and can be displayed on screen in the Interactive mode as described above and can be recorded in a separate file in the case of a Batch run. This is one of the most important features to determine whether the rule order is accurate. It also helps to verify if all the appropriate rules apply to the proto words in the control runs.

3.5.4 Singleton

This feature requires a file with the name of the given model on which it is to be run, such as "shawnee2.sng" and consisting of a list of proto words or etymons. On running the file in this mode, the rules in the model are applied to the data and the output is saved in a temporary file by PHONO called "single.out". This file has to be copied immediately after each run because PHONO saves subsequent runs by deleting it and saving the new output in this file. As with all other files in PHONO, "single.out" is a ASCII text file. The output data is organized as follows:

ETYMOM -->	pa:hpesihe:wa
HP_BECOMES_GP:	=> pa:ʔpesihe:wa
S_BECOMES_TH:	=> pa:ʔpeθihe:wa

SHORTENING1: => paʔpeθihe:wa
 SYNCOPE_HV: => paʔpeθhe:wa

Each etymon is labeled and each rule along with the change brought about by the rule is listed. The final descendant form appears on the last line. In cases where none of the rules in the model apply to the proto word, only the etymon is listed. In order to edit this file, any word processor can be used. The data was edited in WordPerfect 6.1 in the study. The editing consists of three stages:

Stage One: All the unnecessary parts of the data were deleted, such as the label "ETYMON", the arrows, the intermediate stages in the derivation. Most of these were done by developing special macros.

Stage Two: The special characters that are not included in the ASCII character set were replaced as tabulated in section 3.2.

Stage Three: The entire file was then converted into a table with each proto word or etymon in one column and each reflex in another column. The columns were then merged with a table consisting of the English glosses and the grammatical categories of the etymons.

At this stage, the data was sorted alphabetically on the reflexes and presented as in the next chapter.

Chapter Four:

Running the Shawnee Model

4.0 Introduction

Having developed the Shawnee Model, we now turn our attention to the second primary objective of the current study: to use the results of this simulation to increase the evidence for the reconstructed forms in the Proto-Algonquian dictionary; to increase the dictionary by adding the singletons which are supported by the Shawnee data; and to refine those reconstructed forms which do not “fit” with the Shawnee data (see section 1.4). Appendices 1 and 2 list the result of running the *p entries in the PA dictionary and the list of singletons, respectively, on the Shawnee Model. This Output was used to search for reflexes in all available Shawnee data, namely, Voegelin (1938), Andrews (1994) and Chrisley (1995). The number of exact matches and partial matches are listed and discussed in the next section. The second section addresses some points on the data. The last section describes a method that can be used in the future for comparing data after running the entire list of Proto-Algonquian reconstructions both from the PA dictionary and the list of singletons.

4.1 The Preliminary Search

The Output consists of 401 *p entries from the PA dictionary (Appendix 1) and 1020 *p entries from the list of singletons (Appendix 2). The Shawnee data from the three sources consists of approximately 600 entries for “p”. The Output was compared with the available Shawnee data and the result of this comparison yielded 63 matches or near matches (30 from

the PA dictionary and 33 from the singletons) listed below. The list of matches excludes recurring roots and word formatives that are present in the Shawnee data. It appears that the Shawnee etymologies could be added to the list of singletons.

In this list, the Output from the PA dictionary can be distinguished from the Output from the singletons by the MD number. Each entry consists of at least two lines. The first line contains the PHONO Output marked by the plus sign, “+”, to indicate the expected Shawnee form. This is followed by the Proto-Algonquian word followed by the grammatical category of each of the PA forms and the English gloss. The MD number of each form appears in the last column only for words from the PA dictionary. The second line contains the Shawnee forms followed by the English gloss. The Shawnee forms, as indicated earlier, match either partially or totally. If the match is only in the root, the Shawnee entry is marked “root” in brackets, if there is a match in more than the root, the entry is marked “partial” and if there is a total match, the entry is marked “total”. For a note on the suffixes, please see section 2.4. The grammatical category of the Shawnee words are included before the English gloss. Note that the suffixes have not been adjusted in the expected Shawnee forms. In the first example below, the TA suffix in the PA form is *-e:wa. The reflex of this suffix in Shawnee is “-a”. But this is a morphological rule rather than a phonological rule and as discussed in section 2.4 above. The main focus of the current study has been phonological rules.

+pakile:wa nipakila	*pakiθe:wa TA I throw him away, divorce him	TA he casts her off	2521 (total)
+paška:piwa paškaapi	*pa:ška:piwa AI he opened his eyes; he has open eyes	AI one-eyed	2493 (Total)
+pekwi pekowali pekwi	*penkwi NI ashes, dust NI dust, ashes, rubbed to dust, worn out, sore.	NI ashes	2654 (Total)
+pemi pemi-	*pemyi NI grease, tallow	NI oil	2646 (Total)
+pemi?θenwi nipem?θeto nipa?pem?θeto (Reduplication)	*pemihθenwi PT I pass the time PT I pass the resolution around	II flow	2619 (Root)
+pemiwele:wa nipemiwela	*pemiwele:wa TA I guide him, carry him along	TA along	2632 (Total)
+pemiweta:wa nipaapemiwetoopē (Reduplication)	*pemiweta:wa PT we observe religious ceremony	PT carry	2633 (Total)
+pemotamwa nipemota	*pemotamwa TI I shoot at it	TI shoot	2639 (Total)
+pemwe:wa nipemwa pemw-	*pemwe:wa TA I shoot at him TA shoot	TA shoot	2640 (Total)
+pi:čθe:wa piičθeeki	*pi:nči7le:wa AI they go in	AI fly in	2736 (Total)
+pi:talwa:na piitalwa	*pi:ntaθwa:na NA bullet sack	NA quiver	2742 (Total)
+po:tawe:wa pootawe	*po:tawe:wa AI he puts wood on the fire	AI fire	2818 (Total)

+po:tawile:wa nipootaala	*po:tawiθe:wa TA I blow on him	TA blow	2822 (Total)
+po:tawitamwa nipootaata ("awi" undergoes contraction to give "a:").	*po:tawitamwa TI I blow on it	TI blow	2821 (Total)
+poʔkonamwa nipoʔkina (See page 68 below)	*po:ɣkwenamwa TI I cracked it open, broke it with hand	TI break	2827 (Total)
+ptame:wa niptama	*peʔtame:wa TA I accidentally bit him	TA by accident	2676 (Total)
+pye:ta:wa nipyeeeto	*pye:ta:wa PT I bring it	PT bring	2868 (Total)
+pye:wa piyeeki	*pye:wa AI they come	AI come	2886 (Total)
+pa:khikomeʔsinwa nipaakhwa	*pa:k-ahi-kome:-hšin-wa TA I pat him, beat as on drum, tap him	AI knock	(Root)
+paʔθene:wa nipaʔθatena nipaʔθeēena	*pa:ʔθ-en-e:wa TI I tear into it TI I break its stomach, round thing	TA crush	(Root)
+paʔki:kwe:wa nipaʔkatepeena	*pa:θk-i:nkwe:-wa TA I uncovered his head	AI open	(Root)
+pašiphike:wa pašiphike	*pašipi-hi-ke:-wa AI he did the stabbing	AI stab	(Total)
+pašiphwe:wa nipašiphwa	*pašip-ahw-e:wa TA I stab him	TA stab	(Total)
+pawiʔsinwa nipawatena	*pawi-hšin-wa TI I shake it	AI shake	(Root)
+pawʔškawiwa pawʔška	*paw-eškawi-wa AI he shakes dew off as his feet brush the grass	AI shake	(Total)

papaweška (Reduplication)	AI he shakes from excitement		
+pe:khowa nipeekhwa nipeekona	*pe:nki-ho-wa TA I dry him by rubbing TI I dried it by rubbing	AI clean	(Root)
+pekwiwi:kwe?šinwa nipekow?šima	*penkwiw-i:nkwe:-hšin-wa TA I threw him in ashes	AI dust	(Root)
+pešikwe:leme:wa hopešikweeleta	*pešikw-e:lem-e:wa TI he thinks it straight	TA straight	(Partial)
+pešikwika:pawiwa pešikwikaapawi	*pešikwi-ka:pawi-wa AI he stands straight	AI straight	(Total)
+pešikwiweta:wa nipešikwto	*pešikwi-weta:-wa PT I made it level, even,	PT straight straight	(Partial)
+petekwowa nipetekona	*petekw-o-wa TA I roll it around	AI double	(Partial)
+pi:ta:kan piitaaka	*pi:nt-a:kan-i NA sack, bag	NI pocket	(Total)
+pi?tawi?θenwi pi?taw?θenwi ("i" is deleted in the unstressed syllable.)	*pi:htawi-hθen-wi II it lies doubled	II extra layer	(Total)
+po?kitepehwe:wa nipo?kitepe?θeto	*po:xki-tempe:-hw-e:wa TI I broke the head of it	TA open	(Partial)
+po?koθowa nipo?kiθa	*po:xkw-eso-wa TI I make it explode	AI break	(Partial)
+pche:wa nipčikona nipčima	*pe?či-h-e:wa TA I accidentally swallowed him TA I took him for someone else and so spoke to him	TA by accident	(Root)
+pkene:wa nipkama	*paθk-en-e:wa TA I bite off a piece of animate food	TA remove	(Root)

nipkina	TI I pulled it apart by hand, I broke it	(Partial)
+pkha:te:wi nipkaʔhwa	*paθk-ah-a:te:-wi TI remove TA I cut into him, I cut him in two	(Root)
+pkočeʔšinwa pkoče	*pexkw-ečye:-hšin-wa AI lie AI he has a lumpy waist	(Partial)
+pkonamwa nipkona	*peθkw-en-amwa TI open TI I pulled it out from being stuck	(Total)
+pye:čiwe:wa keʔpiyeečiweepθe	*pye:či-h-iwe:-wa AI hither AI you will start going	(Root)
+pye:či:wa nipyecθaha nipyecīha	*pye:či-i:-wa AI hither TA I imported him TA I made him come	(Root)
+pye:čikta:wa hopiyeečineeme teetepθeki	*pye:či-ki-hta:-wa PT forth TI he sees the wagon coming	(Root)

The following matches were pointed out by one of the reviewers:

+pakšimowa taʔpakšimoči	*pankihšimowa AI sunset place where the sun goes, the west	2528
+pakšinwa pakšinwa	*pankihšinwa AI fall AI he alighted from flying	2529
+peθetamwa nipeθetawa	*pesentamwa TI listen TA I pay attention to, listen to, mind him	2685
+pi:menaʔkwa:ni piiminʔkwa	*pyi:menahkwa:ni NI rope NI sewing thread	2889
+pi:leθiwa nipiileθipe	*pi:lesiwa AI clean AI we have an even score; we are revenged	2726
+pi:nʔhamwa nipiinʔha	*pi:nθahamwa TI put in TI I stuck it in	2760

+pke:wa pkweeki	*paθke:wa AI they went off together in another direction	AI separate	2561
+pke:wele:wa nipkeewela	*paθke:wele:wa TA I take him away from the herd	TA withdraw	2562
+po:namwa hopoona	*po:namwa TI he has it	TI lay down	2799
+po:ne:wa hopoonaali	*po:ne:wa TA he has him	TA put	2803
+ppkiwa hapki	*paxpaxkiwa NA prairie chicken	NA partridge	2571
+pye:ta?θenwi piyetθenwi	*pye:ta:θenwi II it smells, smell is wafted this way by wind	II blow hither	2867
+pakθenamwa nipakθena	*paki-hθ-en-amwa TI I turn it loose, I create it	TI let go	
+paleθhe:wa nipaleθha	*paθ-esi-h-e:wa TA I beat her	TA miss	
+paleθiwa paleθi	*paθ-esi-wa AI he misses out	AI miss	
+petekenamwa nipetekina	*petek-en-amwa TI I took it back	TI back	
+pi?taw?škawe:wa nipi?toškawaaki	*pi:htaw-eškaw-e:wa TA I bother them, get in between them	TA extra layer	
+pkwe:ne:wa nipkweena	*paθkwe:-n-e:wa TI I break it off once	TA remove	
+pkwe?kotamwa nipkwe?kota	*paθkwe:-hkwet-amwa TI I cut a piece off it	TI cut	
+pyeptawiwa piyepeto	*pye:h-pahtawi-wa AI he comes running	AI hither	

4.1.1 Results and Discussion

The first point to be noted is that once projections were made for the Shawnee forms, it was much easier to search for the reflexes. In regular reflex search, one has to make the projections for the reflexes mentally. Then, based on these mental projections, the corresponding reflexes can be sought. The task is highly simplified once the projections are available and made into a list to aid the search. Using this method enables us to retain the traditional or classical method of searching for reflexes while having constant access to a point of reference, the Shawnee projections.

However, there are some shortcomings; as already stated, special adjustments have to be made for morphological changes or irregular changes. These are present in this list; for example:

(1) +poʔkonamwa	*po:ɣkwenamwa	TI break	2827
nipoʔkina	TI I cracked it open, broke it with hand		

Post-consonantal “w” is unstable. In the above example, “we” changing to “i” is an irregularity. *w is lost, then *e > i / k_ ; rule 16 in section 2.2.6.

(2) +pawʔškawiwa	*paw-eškawi-wa	AI shake
pawʔška	AI he shakes dew off as his feet brush the grass	
papaweška	AI he shakes from excitement	

There is no syncope in “papaweška”; evidence that syncope operates only on unstressed syllables. Reduplication causes the focus segment to be stressed. Also, syncope does not affect two consecutive syllables:

+pkha:te:wi	*paθk-ah-a:te:-wi	II remove
nipkaʔhwa	TA I cut into him, I cut him in two	

4.1.2 Modifications to the Shawnee Model

There is, of course, room for minor modifications to the Model. Some of the rules that were written into the Model need to be written more precisely. For example, when the Model changes PA “*po:xkwahamwa” (break) to “+poʔkwʔhamwa”, it is unknown whether the vowel following *w will undergo syncope before *h. It is unlikely that this vowel will undergo syncope. In any case, the syncope rule has to be written in such a way so as to exclude such glide plus vowel combinations.

4.2 The Shawnee data and the PA data

The first major problem with the Shawnee data is that none of the three sources that are available is a dictionary. Chrisley’s *An Introduction to the Shawnee Language* is aimed at teaching basic proficiency in the language. It contains a 52-page English to Shawnee word list with less than 800 entries. The most important feature of this work is the recognition by the author of the distinction between phonetic and phonemic transcription. At the beginning of the word list appears the following statement: “NOTE: the symbol [U] is used to show

the word has an unvoiced consonant and suppressed [ʔ] as described on p. 4.” (Chrisley 1995: 27). The status of the glottal stop in many words is questionable. Voegelin (1938) has recorded *mškwí* “blood” and *nim šskomi* “my blood”. And in Chrisley (1995) *mškwí* is recorded with the symbol [ʔ] whereas *nim šskomi* appears as in Voegelin. This indicates that the glottal stop in *nim šskomi* is probably phonetic. There are also some instances in which the glottal stop of a regular reflex is deleted:

*mahkate:we:siwa	AI black	1527
mkateeweθi	he’s black (Chrisley 1995: 31)	
*maθkwa	NA bear	1635
mkwa	bear (Chrisley 1995: 30)	

Both these examples are marked with the symbol “[ʔ]” by Chrisley. The curious “unvoiced” consonants need further investigation. As a result, the glottal stop was the cause of much confusion in the current study. Some of this confusion has been resolved. However, field work needs to be done to compile a more comprehensive dictionary and to resolve some phonological issues.

The PA data contains recurring roots; for example the roots *pi:nt- (inside, inferiority) recurs 38 times in the singletons and 20 times in the PA dictionary. These recurring roots appear in different combinations with several different medials and/or finals. In the Shawnee data, there are only 4 words with the reflex of the root “pi:t”. This situation, in which a given root is attested, a given medial is attested, and a given final is attested, but a given combination of root, medial and final is not attested, or rather simply not yet recorded in the Shawnee

sources, occurs frequently.

The conclusion points to the importance of recording the word formatives and the distribution of the word formatives among the different Algonquian languages. Many of the singletons are innovations in a given language and Shawnee has its own set of innovations. Shawnee has innovated like the other languages using the same roots with different medials and finals. What has been found is that Shawnee has many formatives which are partially cognate but not fully cognate. Shawnee has lots of singletons which are apparent but which are not totally cognate with any of the items in the other languages.

4.3 Method for comparing the data

The first step would be to have as much Shawnee data as possible in a word processor or a database. The product of running the entire PA dictionary and the singletons on the Shawnee Model would be an output of about 10,000 entries. The contents of this list could be compared manually. The other option would be to merge the Shawnee data and the output in a single list and sort it alphabetically. This would result in placing the reflexes very close to the projected reflexes, if not next to them. Needless to say, there would be some problems because of even a single character; for example, the reduplicated Shawnee word “papaweška” would not be placed near “+pawškawiwa”. Thus a combination of a search in a merged list and a manual search in separate lists would provide the optimum result.

Chapter Five: Conclusion

Comparative reconstruction is not an exact science. As a result, many problems become apparent when one tries to develop a computer model to simulate the process of diachronic change. To a linguist, it is obvious as to why a given rule should not apply to a given segment or syllable but such judgements have to be explicitly written into the model in order to permit the computer to resolve each and every single detail. Traditionally, in historical linguistics, rules have been written in a notation which requires some explanatory comments which are tagged along with the rules. Let us take an example from Shawnee which has been previously discussed in section 2.4. If a linguist is given two PA forms one of which contains a first person pronominal prefix, and the other contains a root which begins with “ne”, the traditional rule tells the linguist to apply the rule in the first instance and not apply (elsewhere) in the second instance. For the computer model to make such a judgement on the morphological value of segments or syllables, it would be necessary for the model to parse the form into its components, namely (in the case of Algonquian languages), prefixes, roots, medials, finals, etc. Once a form has been parsed, a whole variety of morphological changes can be specified in the form of rules. Entire paradigms could be stored separately and the model would simply pick appropriate forms and rules. As emphasized throughout, such features are not currently available in PHONO.

It has long been known that if diachronic sound change is regular, then it must be possible

to demonstrate the regularity of sound change in computer models. But very few have actually ventured to take historical sound change rules from text books of well studied languages and develop a working computer model. And anyone who HAS ventured into this territory has quickly realized that there is a world of difference between the rules as they are written in standard linguistic notation and as they need to be written in computer models.

PHONO is the first attempt to develop a standard software for numerous (if not all) language families. What has become very apparent in this study is that a certain amount of morphology has to be dealt with even when we are trying to carry out a phonological study. It would be quite counter-productive to write a whole range of morphological rules which match the change of PA suffixes to Shawnee suffixes based on the verbal paradigms. The primary reason for this is that if we did carry out such an exercise, we would in effect be writing morphological rules as if they were phonological rules.

Thus far, the current study has demonstrated that PHONO can be used for research on the Algonquian language family. As the study continues, more Shawnee forms are being added as evidence for the PA reconstructions.

For further work on the Algonquian language family, more models can be developed. Now that the fundamental mode of operation of PHONO has been realized, and one reasonably good working model has been produced, all subsequent models may be produced rapidly.

The determining factor in producing all subsequent models is the availability of the historical phonology of each of the languages. For many of the languages, quite extensive work in historical phonology has already been done, for example Arapaho and Atsina (Goddard 1974), Munsee (Goddard 1982), Micmac (Hewson 1973b), etc.

In setting up models in PHONO, some other points become quite apparent: PHONO could benefit linguists in teaching Historical Linguistics and Phonology at the undergraduate level.

5.1 Further development of PHONO

Currently, PHONO operates on DOS and requires a very limited amount of disk space. It could be incorporated into a bigger program, preferably running on X-Windows. This program could invoke a relational database, such as FoxPro and other programs. PHONO could even be incorporated into a database. This would make it more user-friendly. It would also permit the user, unlike now, to perform several tasks without having to exit PHONO and run another program and then return to PHONO for further operations.

Also, it has to be modified so that rules can be written in traditional linguistic notation. As it stands, the notation for writing the rules requires some background knowledge of computers and programming.

5.2 Other Applications of PHONO

5.2.1 Research Applications for Other Projects

One application of PHONO in historical linguistics is to use it as demonstrated in this study for languages in which linguists are involved in active field work. For example, the reflexes of Proto-Bantu in the numerous Bantu languages have yet to be recorded. This process may be facilitated if, after some basic preliminary changes have been established, we provide the informants with a list of simulated words, so that the initial hypotheses can be tested, and problems pinpointed.

Word lists are normally collected from the various Bantu languages by means of providing the informants with a list of English and Swahili words. The informants are then requested to give the relevant word in their language. It should be apparent that if the informant were subsequently to be provided with an additional list of expected reflexes from the Proto-language, further work would be greatly facilitated. From a practical point of view, even if the simulated words are inaccurate because of deficiencies in the Makeup and Order of the Model in PHONO, they should help the linguist and the informant arrive at the relevant reflex.

5.2.2 Teaching the Regularity of Sound Change

1. Teaching Historical Linguistics at the undergraduate level: PHONO is the ideal program to teach the regularity of sound change. Currently, historical linguistics is taught with

numerous exercises in phonological changes. The student is given assignments to derive reflexes in daughter languages; such as Greek, Sanskrit, Latin and Proto-Germanic from Proto-Indo-European. Instead of these traditional exercises, PHONO could be used to write the rules and check derivations. The students would learn much more quickly the purpose of doing the exercise, because of the practicality of the program and its interactive nature.

2. Teaching Phonology at the undergraduate level: In general phonology, again, the students are given numerous assignments on phonological rules. The students are given data and are required to devise phonological rules to explain superficial differences in the data. Instead of writing the rules and discussing them in essays, they could write the rules in PHONO. By working on them interactively, they could refine their rules until they arrive at the correct one.

Bibliography

- Andrews, Kenneth Ralph 1994. *Shawnee Grammar*. PhD Dissertation, University of South Carolina.
- Aubin, George F. 1972. *A Historical Phonology of Narragansett*. PhD Dissertation, Brown University. Providence, Rhode Island. (MicroFilm.)
- . 1975. *A Proto-Algonquian Dictionary*. National Museum of Man, Mercury Series, Canadian Ethnology Service Paper 29. Ottawa: National Museums of Canada.
- Bloomfield, Leonard 1925. "On the Sound-system of Central Algonquian." *Language* 1:130-156.
- . 1933. *Language*. New York: Holt, Rinehart and Winston.
- . 1946. "Algonquian." *Linguistic Structures of Native America*. Harry Hoijer et. al., Viking Fund Publications in Anthropology 6. New York: Wenner-Gren Foundation for Anthropological Research:85-129.
- Callender, Charles 1978. "Shawnee." *Handbook of North American Indians*. William C. Sturtevant, (ed.) Washington: Smithsonian Institution, 15:622-635.
- Chrisley, Ronald L. 1995. *An Introduction to the Shawnee Language*. (Published privately.)
- Goddard, Ives 1965. "The Eastern Algonquian Intrusive Nasal." *IJAL* 31:206-20.
- . 1967a. "Notes on the Genetic Classification of the Algonquian Languages." *National Museum of Canada, Bulletin* 214:7-12.
- . 1967b. "The Algonquian Independent Indicative." *National Museum of Canada, Bulletin* 214:66-106.
- . 1974. "An Outline of the Historical Phonology of Arapaho and Atsina." *IJAL* 40:102-16.
- . 1978. "Eastern Algonquian Languages." *Handbook of North American Indians*. William C. Sturtevant, (ed.) Washington: Smithsonian Institution, 15:70-77.

- . 1982. "The Historical Phonology of Munsee." *IJAL* 48:16-48.
- Hall, Robert A. Jr. 1983. *Proto-Romance Morphology*. Philadelphia: John Benjamins.
- Handbook of North American Indians: Northeast. (Vol. 15)* 1978. William C. Sturtevant, (ed.) Washington: Smithsonian Institution.
- Hartman, Steven Lee 1981. "A Universal Alphabet for Experiments in Comparative Phonology." *Computers and the Humanities* 15:75-82.
- . 1993. "Writing Rules for a Computer Model of Sound Change." *Southern Illinois Working Papers in Linguistics and Language Teaching* 2:31-39.
- Hewson, John 1973a. "Towards a Proto-Algonkian Dictionary." *Studies in Linguistics* 23:63-68.
- . 1973b. "Proto-Algonkian Reflexes in Micmac." *Anthropological Linguistics* 15:151-64.
- . 1974. "Comparative Reconstruction on the Computer." *Proceedings of the First International Conference on Historical Linguistics*. Ed. by J. M. Anderson and C. Jones, 191-97. Amsterdam: North Holland.
- . 1977. *Reconstructing Prehistoric Languages on the Computer: The Triumph of the Electronic Neogrammarian*. Computational and Mathematical Linguistics: Proceedings of the International Conference on Computational Linguistics [Pisa, 1973]:263-273. Florence: Leo S. Olschki Editore.
- . 1991. "Verbal Derivation in Micmac." *Journal of the Atlantic Provinces Linguistic Association* 13:21-33.
- . 1993. *A Computer-Generated Dictionary of Proto-Algonquian*. Ottawa: Canadian Museum of Civilization.
- Hock, Hans Henrich 1991. *Principles of Historical Linguistics*. (Second edition.) New York: Mouton de Gruyter.
- Hockett, Charles F. 1957. "Central Algonquian Vocabulary: stems in /k-/." *IJAL* 23:247-68.
- International Encyclopedia of Linguistics*. 1992. William Bright (editor-in-chief). New

York: Oxford University Press.

Michelson, Truman 1914. "Algonquian Linguistic Miscellany." *Journal of the Washington Academy of Sciences* 4:402-9.

———. 1917. "Notes on Algonquian Languages." *IJAL* 1:50-7.

———. 1920. "Two Phonetic Shifts Occurring in Many Algonquian Languages." *IJAL* 1:300-4.

———. 1935. "Phonetic Shifts in Algonquian Languages." *IJAL* 8:131-71.

Miller, Wick R. 1959. "An Outline of Shawnee Historical Phonology." *IJAL* 25:16-21.

Pentland, David H. 1979. *Algonquian Historical Phonology*. PhD Dissertation, University of Toronto.

——— and H. Christoph Wolfart 1982. *Bibliography of Algonquian Linguistics*. Winnipeg: University of Manitoba Press.

Siebert, Frank T. Jr. 1941. "Certain Proto-Algonquian Consonant Clusters." *Language* 17:298-303.

———. 1967. "The Original Home of the Proto-Algonquian people." *National Museum of Canada, Bulletin* 214:13-47.

———. 1975. "Resurrecting Virginia Algonquian from the dead: the reconstituted and historical phonology of Powhatan." *Studies in Southeastern Indian Languages*. Ed. by James M. Crawford, 285-453. Athens: University of Georgia Press.

Voegelin, Carl F. 1935. "Shawnee Phonemes." *Language* 11: 23-37.

———. 1936. "Productive Paradigms in Shawnee." *Essays in anthropology presented to A. L. Kroeber*. Ed. by Robert H. Lowie, 391-403. Berkeley: University of California Press.

———. 1938. *Shawnee Stems and the Jacob P. Dunn Miami Dictionary*. Indianapolis: Indiana Historical Society.

Appendix 1

The following is a list of the Proto-Algonquian *p entries in the PA dictionary run on the Shawnee Model. Each entry consists of the PHONO Output marked by the plus sign, "+", in the first column to indicate the expected Shawnee form. This list was sorted on these expected forms. (Note: "θ" is sorted as "d"). The second column contains the Proto-Algonquian word followed by the grammatical category of each of the PA forms and the English gloss. The MD number of each form appears in the last column.

+pa:ka?kwe:wa	*pa:ka:xkwe:wa	AI he bumps into tree	2477
+pa:ka?kw?hike:wa	*pa:ka:xkwahike:wa	AI beat	2476
+pa:kšiwa	*pa:kihšiwa	AI swell	2478
+pa:kwe:wi	*pa:kwe:wi	II it is shallow	2479
+pa:kwi?θenwi	*pa:kwihθenwi	II ground	2480
+pa:limawe	*pa:θimawe	XP later	2500
+pa:pek-	*pa:pek-	XP immediately	2481
+pa:pye?ta:wa	*pa:pye:hta:wa	PT wait	2482
+pa:šiči	*pa:šiči	PV over	2489
+pa:šithamwa	*pa:šithamwa	TI step	2490
+pa:šithwe:wa	*pa:šithawe:wa	TA step	2491
+pa:we?tekwi	*pa:we?tekwi	NI rapids	2506
+paθekwi:wa	*pasekwi:wa	AI he arises from sitting	2542
+paθete:wi	*pasete:wi	II hot	2543
+paθhamwa	*pa:ʔθahamwa	TI crush	2488
+paka:n-	*paka:n-	N nut	2513
+pakama:kani	*pakama:kani	NI cudgel	2514
+pakame:wa	*pakame:wa	TA he strikes him	2515
+pakatamwa	*pakantamwa	TI he strikes it	2517
+pakθenwi	*pankihθenwi	II fall	2530
+pake:wa	*pake:wa	AI he pounds, he strikes	2518
+pakhanwi	*pankahanwi	II pulse	2525
+pakhokowa	*pankakhokowa	AI pulse	2526
+pakile:wa	*pakiθe:wa	TA he casts her off	2521
+pakitamwa	*pakitamwa	TI throw	2519
+pakitene:wa	*pakitene:wa	TA set down	2520
+pakšime:wa	*pankihšime:wa	TA drop	2527
+pakšimowa	*pankihšimowa	AI sunset	2528
+pakšinwa	*pankihšinwa	AI fall	2529
+pawkane:ya:wi	*pawkane:ya:wi	II perforated	2523
+pala:čihe:wa	*paθa:čihe:wa	TA ruin	2549
+pala:čta:wa	*paθa:čihta:wa	PT ruin	2550
+pala:pame:wa	*paθa:pame:wa	TA lose	2551
+pala:patamwa	*paθa:pantamwa	TI lose	2552

+pala:teθiwa	*paθa:tesiwa	AI ruin	2553
+palame?kwa	*paθameθkwa	NA young beaver	2556
+pale:ne?kwi	*pale:neθkwi	XP five	2524
+palenamwa	*paθenamwa	TI drop	2557
+palene:wa	*paθene:wa	TA drop	2558
+palhwe:wa	*paθahwe:wa	TA miss	2555
+pal?hamwa	*paθahamwa	TI miss	2554
+papa:ma:teθiwa	*papa:ma:tesiwa	AI live	2533
+papa:ma:teθiweni	*papa:ma:tesiweni	NI travel	2534
+papa:ma:θenwi	*papa:ma:θenwi	II blown	2532
+papa:ma:θiwa	*papa:ma:θiwa	AI blown	2531
+papa:me:leme:wa	*papa:me:leme:wa	TA attention	2535
+papa:mi	*papa:mi-	PV about	2537
+papa:mipatawiwa	*papa:mipatawiwa	AI run	2538
+papa:mi?θe:wa	*papa:mi?le:wa	AI fly	2539
+papa:miwele:wa	*papa:miwele:wa	TA carry	2540
+papa:m?ška:wa	*papa:meška:wa	AI go around	2536
+papa:mwo?θe:wa	*papa:mwohθe:wa	AI walk	2541
+papta:wa	*pa:hpihta:wa	PT laugh	2474
+pa?θamwa	*pa:θamwa	TI dry	2483
+pa?θowa	*pa:θowa	AI dry	2484
+pa?θwe:wa	*pa:θwe:wa	TA dry	2485
+pa?kenamawe:wa	*pa:θkenamawe:wa	TA open	2501
+pa?kenamwa	*pa:θkenamwa	TI open	2502
+pa?kene:wa	*pa:θkene:wa	TA open	2503
+pa?ketonwa	*pa:θketonwa	AI open	2504
+pa?kwi:kwe:le:wa	*pa:θkwi:nkwe:θe:wa	TA uncover	2505
+pa?piwa	*pa:hpiwa	AI laugh	2475
+pa?tawime:wa	*pa:θtawime:wa	TA accuse	2486
+pa?te:wi	*pa:θte:wi	II dry	2487
+paška:phwe:wa	*pa:ška:pahwe:wa	TA burst	2492
+paška:piwa	*pa:ška:piwa	AI one-eyed	2493
+paška:wehowa	*pa:ška:we:howa	AI hatch	2494
+paškeθamwa	*pa:škeamwa	TI shoot	2496
+paškeθikani	*pa:škesikani	NI gun	2497
+paškeθike:wa	*pa:škesike:wa	AI shoot	2498
+paškeθwe:wa	*pa:škeswe:wa	TA shoot	2499
+paškhamwa	*pa:škahamwa	TI burst	2495
+patkecehwe:wa	*patahkecehwe:wa	TA pierce	2548
+patkhamwa	*patahkahamwa	TI pierce	2546
+patkhwe:wa	*patahkahwe:wa	TA prick	2547

+pawa:miwenwi	*pawa:miwenwi	NI dream	2566
+pawawile:wa	*pawawiθe:wa	TA dream	2570
+pawawitamwa	*pawawitamwa	TI dream	2569
+paw?hamwa	*pawahamwa	TA beat	2567
+paw?hikani	*pawahikani	NI thresh	2568
+pčipowa	*pe?čipowa	AI poison	2663
+pčipoweni	*pe?čipoweni	NI poison	2664
+pčito:tamwa	*pe?čito:tamwa	TI error	2665
+pčta:wa	*pe?čih̄ta:wa	PT accidental	2662
+pθ	*pe?θ-	PV in small particles	2683
+pθakw	*pahsakh-	PV sticky, muddy, miry	2511
+pθkamwa	*pe?sehkamwa	TI by accident	2666
+pθkawe:wa	*pe?sehkawe:wa	TA hit	2667
+pe:čikiwa	*pe:čikiwa	AI grow slowly	2573
+pe:θka:wa	*pe:sehka:wa	AI slow	2577
+pe:θka:wi	*pe:sehka:wi	II go slow	2578
+pe:koθiwa	*pe:nkwesiwa	AI clean	2574
+pe:tamwa	*pe:ntamwa	TI hear	2575
+pe:tawe:wa	*pe:ntawe:wa	TA hear	2576
+pe:two?θe:wa	*pe:twoh̄θe:wa	AI slowly	2579
+peθeniwa	*pensenliwa	AI he has foreign body in eye	2655
+peθetamwa	*pesentamwa	TI listen	2685
+pekiwa	*pekiwa	NA resin	2581
+pekša	*penkwehša	NA fly	2653
+pekwa	*penkwa	NA buffalo gnat, black fly	2651
+pekwi	*penkwi	NI ashes	2654
+pel	*pel-	PV strip, tear off	2582
+pele:nikwa	*pele:nikwa	NA flying squirrel	2584
+pele:wa	*pele:wa	NA partridge	2585
+pele?šya?ša	*pele:hšyahša	NA bird	2583
+pema:cheθowa	*pema:čih̄esowa	AI live	2588
+pema:che:wa	*pema:čihe:wa	TA life	2587
+pema:chiwe:wa	*pema:čihiwe:wa	AI life	2590
+pema:čta:wa	*pema:čihta:wa	PT life	2591
+pema:mo:wa	*pema:mo:wa	AI flee	2594
+pema:powete:wi	*pema:powete:wi	II float	2595
+pema:taka:wa	*pema:taka:wa	AI along	2598
+pema:teθiwa	*pema:tesiwa	AI live	2599
+pema:teθiweni	*pema:tesiweni	NI life	2600
+pema:waletiwaki	*pema:waletiwaki	AI file	2601
+pemahokowa	*pema:hwekowa	AI along	2592

+pemahote:wi	*pema:hwete:wi	II float	2593
+pemakočinwa	*pemakočinwa	AI along	2604
+pema7θenwi	*pema:7θenwi	II along	2597
+pema7kwi7θenwi	*pema:xkwihθenwi	II extend	2603
+pema7kwi7šinwa	*pema:xkwihšinwa	AI lengthwise	2602
+pema7šiwa	*pema:7šiwa	AI along	2596
+peme:leme:wa	*peme:leme:wa	TA attention	2605
+peme:letamwa	*peme:lentamwa	TI heed	2606
+pemeke:wa	*pemeke:wa	AI he dances along, by	2607
+pemene:wa	*pemene:wa	TA care for	2609
+pemeniwe:wa	*pemeniwe:wa	AI care	2610
+pemi	*pemyi	NI oil	2646
+pemičθenwi	*pemičihθenwi	II across	2615
+pemiči	*pemiči	XP sideways	2613
+pemičime:wa	*pemičime:wa	AI along	2616
+pemičiwani	*pemičiwani	II flow	2617
+pemičšinwa	*pemičihšinwa	AI across	2614
+pemika:pawiwa	*pemika:pawiwa	AI stand	2620
+pemilešhwe:wa	*pemiθešihwe:wa	TA along	2631
+pemipeθowa	*pemipesowa	AI along	2624
+pemipete:wi	*pemipete:wi	II along	2625
+pemiphe:wa	*pemipahe:wa	TA run	2621
+pemiphiwe:wa	*pemipahiwe:wa	AI flee	2622
+pemipokowa	*pemipokowa	AI float	2626
+pemiptawiwa	*pemipahtawiwa	AI run	2623
+pemi7θe:wa	*pemi7le:wa	AI fly	2627
+pemi7θenwi	*pemi7hθenwi	II flow	2619
+pemi7šinwa	*pemi7hšinwa	AI lie	2618
+pemitapiwa	*pemitapiwa	AI sit	2628
+pemitēce7šinwa	*pemitēce:7hšinwa	AI lie	2630
+pemite:wi	*pemite:wi	NI broth	2629
+pemiwele:wa	*pemiwele:wa	TA along	2632
+pemiweta:wa	*pemiweta:wa	PT carry	2633
+pemočike:wa	*pemočike:wa	AI missile	2638
+pemo:me:wa	*pemo:me:wa	TA carry	2634
+pemo:mekowa	*pemo:mekowa	AI ride	2635
+pemo:tamwa	*pemo:ntamwa	TI carry	2636
+pemo:te:wa	*pemo:te:wa	AI crawl	2637
+pemotamwa	*pemotamwa	TI shoot	2639
+pem7ška:wa	*pemeška:wa	AI along	2611
+pem7ška:wi	*pemeška:wi	II along	2612

+pemwe:wa	*pemwe:wa	TA shoot	2640
+pemwe:we?θenwi	*pemwe:we:hθenwi	II along	2642
+pemwe:we?šinwa	*pemwe:we:hšinwa	AI go	2641
+pemwoθhe:wa	*pemwohθahe:wa	TA walk	2643
+pemwo?θatamwa	*pemwohθatamwa	TI walk	2644
+pemwo?θe:wa	*pemwohθe:wa	AI walk	2645
+penenamwa	*penenamwa	TI down	2648
+penene:wa	*penene:wa	TA down	2649
+peni?šime:wa	*penihšime:wa	TA down	2650
+pen?ha:kani	*penaha:kani	NI comb	2647
+pepikwa	*pepikwa	NA flea	2656
+pepo:ni?šiwa	*pepo:ni:hšiwa	AI winter	2658
+pepo:no:wi	*pepo:no:wi	II it is winter	2659
+pepo:nwi	*pepo:nwi	II winter	2661
+pešiwa?tekwa	*pešiwa:htekwa	NA dogwood	2693
+peškiwa	*pešekiwa	NA buffalo	2691
+pi:čθe:wa	*pi:nči?le:wa	AI fly in	2736
+pi:čθenwi	*pi:nčihθenwi	II inside	2734
+pi:čipete:wi	*pi:nčipete:wi	II fly in	2735
+pi:čšinwa	*pi:nčihšinwa	AI inside	2733
+pi:θkamwa	*pi:nšekamwa	TI put on	2737
+pi:θowa	*pi:nsowa	AI he is enclosed	2738
+pi:koče?θakw	*pi:kwečye:hθakw-	N rotten	2710
+pi:konamwa	*pi:kwenamwa	TI break	2711
+pi:kone:wa	*pi:kwene:wa	TA break	2712
+pi:kška:wa	*pi:kweška:wa	AI break	2713
+pi:kška:wi	*pi:kweška:wi	II break	2714
+pi:kškamwa	*pi:kweškamwa	TI break	2715
+pi:kškawe:wa	*pi:kweškawē:wa	TA break	2716
+pi:kwakana.me:wa	*pi:kwakana.me:wa	TA break	2706
+pi:kwakana:tamwa	*pi:kwakana:ntamwa	TI break	2707
+pi:kwame:wa	*pi:kwame:wa	TA break	2708
+pi:kwa?θenwi	*pi:kwa:θenwi	II blow	2702
+pi:kwa?šiwa	*pi:kwa:šiwa	AI blow	2701
+pi:kwatamwa	*pi:kwantamwa	TI break	2709
+pi:kwhwe:wa	*pi:kwahwe:wa	TA break	2705
+pi:kwipečikani	*pi:kwipečikani	NI plow	2720
+pi:kwipečike:wa	*pi:kwipečike:wa	AI plough	2721
+pi:kwipele:wa	*pi:kwipeθe:wa	TA tear	2723
+pi:kwipetamwa	*pi:kwipetamwa	TI tear	2722
+pi:kwi?θenwi	*pi:kwihθenwi	II break	2719

+pi:kwí?šime:wa	*pi:kwihšime:wa	TA break	2717
+pi:kwí?šinwa	*pi:kwihšinwa	AI break	2718
+pi:kw?hamwa	*pi:kwahamwa	TI smash	2703
+pi:kw?hike:wa	*pi:kwahike:wa	AI break	2704
+pi:la:kamiwi	*pi:la:kamiwi	II clean	2724
+pi:latwi	*pi:latwi	II clean	2725
+pi:leθiwa	*pi:lesiwa	AI clean	2726
+pi:lina:koθiwa	*pi:lina:kwesiwa	AI clean	2731
+pi:lina:kwati	*pi:lina:kwatwi	II clean	2730
+pi:li?čike:wa	*pi:lihčike:wa	AI clean	2727
+pi:li?ta:wa	*pi:lihta:wa	PT clean	2729
+pi:loθo:wiwa	*pi:loso:wiwa	AI clean	2732
+pi:l?he:wa	*pi:lihe:wa	TA clean	2728
+pi:menamwa	*pyi:menamwa	TI twist	2890
+pi:mena?kwa:ni	*pyi:menahkwa:ni	NI rope	2889
+pi:menikani	*pyi:menikani	NI screw	2892
+pi:mikwayapele:wa	*pyi:mikwayapeθe:wa	TA wring	2893
+pi:mikwayawene:wa	*pyi:mikwayawene:wa	TA twist	2894
+pi:molowa	*pyi:mwelowa	AI carry	2895
+pi:m?hikani	*pyi:mahikani	NI screw	2888
+pi:nhwe:wa	*pi:nθahwe:wa	TA put in	2761
+pi:n?hamwa	*pi:nθahamwa	TI put in	2760
+pi:talwa:na	*pi:ntaθwa:na	NA quiver	2742
+pi:te:wi	*pi:nte:wi	II inside	2743
+pi:tenamwa	*pi:ntenamwa	TI put in	2744
+pi:tene:wa	*pi:ntene:wa	TA put in	2745
+pi:thwe:wa	*pi:ntahwe:wa	TA bury	2740
+pi:tikale:wa	*pi:ntwikale:wa	TA inside	2747
+pi:tikameki	*pi:ntwikamenki	XP indoors	2748
+pi:tikata:wa	*pi:ntwikata:wa	PT bring	2749
+pi:tikatawe:wa	*pi:ntwikatawe:wa	TA inside	2750
+pi:tikawe:wa	*pi:ntwikawe:wa	TA inside	2751
+pi:tike	*pi:ntwike	XP inside	2752
+pi:tike:lešhwe:wa	*pi:ntwike:θešihwe:wa	TA inside	2757
+pi:tike:peθowa	*pi:ntwike:pesowa	AI inside	2755
+pi:tike:phe:wa	*pi:ntwike:pahe:wa	TA inside	2753
+pi:tike:ptawiwa	*pi:ntwike:pahtawiwa	AI inside	2754
+pi:tike:wa	*pi:ntwike:wa	AI enter	2758
+pi:tike:we:penamwa	*pi:ntwike:we:penamwa	TI fling in	2759
+pi:tike?θe:wa	*pi:ntwike:ʔle:wa	AI fly in	2756
+pi:tkate:wa:ni	*pi:ntaθkate:wa:ni	NI powder horn	2741

+pi:tolak	*pi:ntoθak-	XP inside	2746
+pi:wa:pkwak kwa	*pi:wa:peθkwaxkeh kwa	NA kettle	2782
+pi:wa:pkwi	*pi:wa:peθkwi	NI metal	2783
+pi:wačikani	*pi:wancikani	NI crumb	2784
+pi:wačike:wa	*pi:wancike:wa	AI crumb	2785
+pi:wanwi	*pi:wanwi	II snow	2786
+pi:wapiwa	*pi:wapiwa	AI pieces	2787
+pi:wa?te:wi	*pi:wa?te:wi	II scattered	2788
+pi:wene:wa	*pi:wene:wa	TA crumble	2791
+pi:we?kole:wa	*pi:wehkweθe:wa	TA pieces	2790
+pi:we?kotamwa	*pi:wehkwetamwa	TI pieces	2789
+pi:wešikani	*pi:wešikani	NI scrap	2792
+pi:wikhikani	*pi:wikhikani	NA chip	2793
+pi:wiwe:pene:wa	*pi:wiwe:pene:wa	TA scatter	2794
+piθenwi	*pi:θihenwi	II to pieces	2768
+piθhamwa	*pi:θahamwa	TI to pieces	2762
+piθhwe:wa	*pi:θahwe:wa	TA to pieces	2763
+piθšinwa	*pi:θihšinwa	AI to pieces	2767
+piθakana:me:wa	*pi:θakana:me:wa	TA to pieces	2764
+piθame:wa	*pi:θame:wa	TA chew up fine	2765
+piθeθiwa	*pi:θesiwa	AI particles	2766
+piθikhamwa	*pi:θikhamwa	TI chop	2769
+piθikhwe:wa	*pi:θikahwe:wa	TA chop	2770
+piθipele:wa	*pi:θipele:wa	TA to pieces	2771
+piθipočikani	*pi:θipwečikani	NI mill	2772
+piθipočike:wa	*pi:θipwečike:wa	AI grind	2773
+piθipoθowa	*pi:θipwesowa	AI grind	2775
+piθipole:wa	*pi:θipwele:wa	TA grind	2774
+piθipota:wa	*pi:θipweta:wa	PT grind	2776
+piθipote:wi	*pi:θipwete:wi	II grind	2777
+piθya:wi	*pi:θya:wi	II fine	2778
+pi?tawikenwi	*pi:htawikenwi	II extra layer	2700
+pi?te:w-	*pi:te:w-	N foam	2779
+piš kwa	*pi:š kwa	NA nighthawk	2780
+pkate:wa	*pahkate:wa	AI hungry	2508
+pke:wa	*paθke:wa	AI separate	2561
+pke:wele:wa	*paθke:wele:wa	TA withdraw	2562
+pkelawe:wa	*paθkelawe:wa	TA beat	2563
+pkelawike:wa	*paθkelawike:wa	AI win	2564
+pke?kanaw	*paθke:θkanaw-	XP off the road	2560
+pkhamwa	*paθkahamwa	TI sever	2559

+pkipele:wa	*paθkipeθe:wa	TA tear	2565
+pkwa?kwa:wi	*pexkwa:ɣkwa:wi	NI clump	2695
+pkwa?kwati	*pexkwa:ɣkwatwi	NI lump	2696
+pkwa?kwileče?tawe:wa	*pexkwa:ɣkwileθenčye:ʔtawe:wa	TA punch	2697
+pkwe:šikan	*pahkwe:šikan-	N bread	2509
+pkwe:šwe:wa	*pahkwe:šwe:wa	TA cut	2510
+pkwika:pawiwaki	*pexkwika:pawiwaki	AI stand	2698
+pkwikana:kani	*pexkwikana:kani	NI ankle	2699
+po:θhe:wa	*po:sihe:wa	TA load	2810
+po:θiwa	*po:siwa	AI embark	2814
+po:θiwe:pene:wa	*po:siwe:pene:wa	TA throw	2815
+po:θiwenwi	*po:siwenwi	NI embarkation	2816
+po:θta:wa	*po:sihta:wa	PT load	2812
+po:θta?θowa	*po:sihta:ʔsowa	AI load	2811
+po:θtawe:wa	*po:sihtawe:wa	TA load	2813
+po:na:nemati	*po:na:nematwi	II cease	2797
+po:na:nematwi	*po:na:piwa	AI cease	2798
+po:namwa	*po:namwa	TI lay down	2799
+po:ne:leme:wa	*po:ne:leme:wa	TA cease	2800
+po:ne:letamwa	*po:ne:lentamwa	TI cease	2802
+po:ne:wa	*po:ne:wa	TA put	2803
+po:ni:wa	*po:ni:wa	AI stop	2804
+po:nime:wa	*po:nime:wa	TA stop	2807
+po:ni?ta:wa	*po:ni?ta:wa	AI stop	2808
+po:ni?ta:wa	*po:nihta:wa	PT stop	2806
+po:niwe?kamwa	*po:niwehkamwa	TI stop	2809
+po:n?he:wa	*po:nihe:wa	TA stop	2805
+po:tawa:le:wa	*po:tawa:θe:wa	TA fire	2817
+po:tawe:wa	*po:tawe:wa	AI fire	2818
+po:tawičikani	*po:tawičikani	NI blow	2819
+po:tawičike:wa	*po:tawičike:wa	AI blow	2820
+po:tawile:wa	*po:tawiθe:wa	TA blow	2822
+po:tawitamwa	*po:tawitamwa	TI blow	2821
+po:yawe:wa	*po:yawe:wa	NA beaver up to one year of age	2839
+pokškamwa	*po:ɣkweškamwa	TI break	2830
+pokškawē:wa	*po:ɣkweškawē:wa	TA break	2831
+po?kama	*po:θkama	NA wild american plum	2823
+po?konamwa	*po:ɣkwenamwa	TI break	2827
+po?kone:wa	*po:ɣkwene:wa	TA break	2828
+po?košamwa	*po:ɣkwešamwa	TI cut open	2829
+po?kwhwe:wa	*po:ɣkwahwe:wa	TA break	2826

+po?kwika:te:hwe:wa	*po:xkwika:te:hwe:wa	TA break	2834
+po?kwine?kehwe:wa	*po:xkwine?ke:hwe:wa	TA break	2835
+po?kwipole:wa	*po:xkwipwe?e:wa	TA break	2837
+po?kwipota:wa	*po:xkwipweta:wa	PT break	2836
+po?kwi??enwi	*po:xkwi??enwi	II break	2832
+po?kwi??eta:wa	*po:xkwi??eta:wa	PT break	2833
+po?kwitepehwe:wa	*po:xkwitempe:hwe:wa	TA break	2838
+po?kw?hamwa	*po:xkwahamwa	TI break	2825
+po?po?kwa	*po:hpo:hkwa	NA bobwhite, quail	2795
+ppkiwa	*paxpaxkiwa	NA partridge	2571
+pš-	*pehš-	PV peel, husk	2580
+pškale:wa	*peskale:wa	AI blaze	2686
+pškale:wi	*peskale:wi	II blaze	2687
+pškenamwa	*peskenamwa	TI fold	2688
+pškene:wa	*peskene:wa	TA fold	2689
+pškolawe:wa	*peškwelawe:wa	TA miss	2694
+pškone:wa	*paškwene:wa	TA pluck	2544
+pškošwe:wa	*paškwēšwe:wa	TA cut	2545
+pškye:kenamwa	*peskye:kenamwa	TI fold	2690
+ptakana:me:wa	*pe?takana:me:wa	TA by accident	2674
+ptakana:tamwa	*pe?takana:ntamwa	TI by accident	2675
+ptakškamwa	*pe?ta:xkweškamwa	TI by accident	2668
+ptakškawe:wa	*pe?ta:xkweškawē:wa	TA strike	2669
+ptame:wa	*pe?tame:wa	TA by accident	2676
+pta?kwi?šinwa	*pe?ta:xkwihšinwa	AI bump	2670
+ptatamwa	*pe?tantamwa	TI by mistake	2677
+ptelawe:wa	*pe?telawe:wa	TA accidentally	2678
+ptenamawe:wa	*pe?tenamawe:wa	TA take	2679
+ptenamwa	*pe?tenamwa	TI take	2680
+ptene:wa	*pe?tene:wa	TA accidental	2681
+ptešo?owa	*pe?tešwesowa	AI accidental	2682
+pthamwa	*pe?tahamwa	TI accident	2671
+ptho?owa	*pe?tahwesowa	AI accidental	2673
+pthwe:wa	*pe?tahwe:wa	TA by mistake	2672
+pwa:lawi?ta:wa	*pwa:θawihta:wa	PT fail	2841
+pwa:law?he:wa	*pwa:θawihe:wa	TA fail	2840
+pwa:wale:wa	*pwa:wa?e:wa	AI load	2842
+pwa:wo:me:wa	*pwa:wo:me:wa	TA fail	2843
+pwe:kičile:wa	*pwe:kičiče:wa	TA fart	2844
+pwe:kitiwa	*pwe:kitiwa	AI fart	2845
+pye:č?e:wa	*pye:či?le:wa	AI fly hither	2851

+pye:éθe:wi	*pye:ci7le:wi	II fly hither	2852
+pye:cićime:wa	*pye:cićime:wa	AI hither	2846
+pye:cićiwanwi	*pye:cićiwanwi	II current	2847
+pye:ćika:te:wi	*pye:ćika:te:wi	II bring	2848
+pye:ćime:wa	*pye:ćime:wa	TA call hither	2849
+pye:ćimyawekoθiwa	*pye:ćimyawekwesiwa	AI come	2850
+pye:θa:pame:wa	*pye:sa:pame:wa	TA see coming	2857
+pye:θka:wi	*pye:sehka:wi	II come hither	2858
+pye:le:wa	*pye:θe:wa	TA bring	2855
+pye:makati	*pye:makatwi	II come	2856
+pye:ta:ćime:wa	*pye:ta:ćyeme:wa	TA come	2859
+pye:ta:ćimowa	*pye:ta:ćyemowa	AI bring	2860
+pye:ta:mo:wa	*pye:ta:mo:wa	AI flee hither	2861
+pye:ta:nemati	*pye:ta:nematwi	II blow hither	2862
+pye:ta:panwi	*pye:ta:panwi	II dawn	2863
+pye:ta:wa	*pye:ta:wa	PT bring	2868
+pye:takoćinwa	*pye:takoćinwa	AI come	2869
+pye:takote:wi	*pye:takote:wi	II hither	2870
+pye:tamawe:wa	*pye:tamawe:wa	TA bring	2871
+pye:ta7θamapiwa	*pye:ta:7θamapiwa	AI face hither	2865
+pye:ta7θamwo7θe:wa	*pye:ta:7θamwohθe:wa	AI walk hither	2866
+pye:ta7θenwi	*pye:ta:7θenwi	II blow hither	2867
+pye:ta7šiwa	*pye:ta:7šiwa	AI blow hither	2864
+pye:tawe:wa	*pye:tawe:wa	TA bring	2872
+pye:tenamawe:wa	*pye:tenamawe:wa	TA hand to	2873
+pye:tene:wa	*pye:tene:wa	TA hand hither	2874
+pye:tkwe:we:wa	*pye:teθkwe:we:wa	AI bring	2875
+pye:to:me:wa	*pye:to:me:wa	TA bring	2876
+pye:to:tamwa	*pye:to:ntamwa	TI bring	2877
+pye:twe:we:keθiwa	*pye:twe:we:kesiwa	AI come	2880
+pye:twe:we:peθowa	*pye:twe:we:pesowa	AI come	2882
+pye:twe:we:ptawiwa	*pye:twe:we:pahtawiwa	AI come	2881
+pye:twe:we:7θenwi	*pye:twe:we:hθenwi	II come	2879
+pye:twe:we:7šinwa	*pye:twe:we:hšinwa	AI come	2878
+pye:twe:witamwa	*pye:twe:witamwa	TI come	2883
+pye:two7θe:wa	*pye:twohθe:wa	AI walk hither	2884
+pye:wa	*pye:wa	AI come	2886
+pye:he:wa	*pye:he:wa	TA wait	2853
+pyehowa	*pye:howa	AI wait	2854
+pye7ta:wa	*pye:hta:wa	PT wait for	2855

Appendix 2

The following is a list of the *p entries of singletons run on the Shawnee Model. Each entry consists of the PHONO Output marked by the plus sign, “+”, in the first column to indicate the expected Shawnee form. The list was sorted on these expected forms. The second column contains the Proto-Algonquian word followed by the grammatical category of each of the PA forms and the English gloss. Since the singletons have not been incorporated into the MD, no MD number is cited in this list.

+pa:ka?kwi?θenwi	*pa:k-a:ɣkwi-hθen-wi	II	knock
+pa:ka?kwi?θeta:wa	*pa:k-a:ɣkwi-hθeta:-wa	PT	dash
+pa:ka?kwi?šime:wa	*pa:k-a:ɣkwi-hšim-e:wa	TA	dash
+pa:ka?kwi?šinwa	*pa:k-a:ɣkwi-hšin-wa	AI	knock
+pa:khikome?šinwa	*pa:k-ahi-kome:-hšin-wa	AI	knock
+pa:kiθite:paliwa	*pa:ki-site:-pali-wa	AI	swollen
+pa:kitepehikani	*pa:ki-tempe:-h-ik-ani	NI	club
+pa:pemikenwi	*pa:pemi-ken-wi	II	grow
+pa:pemikhe:wa	*pa:pemi-ki-h-e:wa	TA	grow
+pa:pemikiwa	*pa:pemi-ki-wa	AI	grow
+pa:pemikta:wa	*pa:pemi-ki-hta:-wa	PT	grow
+pa:pemipeθowa	*pa:pemi-peso-wa	AI	along
+pa:pemipete:wi	*pa:pemi-pete:-wi	II	along
+pa:pepakešamwa	*pa:pepak-eš-amwa	TI	thin
+pa:pepakešwe:wa	*pa:pepak-ešw-e:wa	TA	thin
+pa:pi:we:kenamwa	*pa:pi:w-e:k-en-amwa	TI	particle
+pa:po?kilečike:wa	*pa:po:ɣki-θenčyi-ke:-wa	AI	break
+pa:pwa?pa:kone?θenwi	*pa:pwap-a:kone:-hθen-wi	II	break
+pa:pwa?pa:kone?šinwa	*pa:pwap-a:kone:-hšin-wa	AI	break
+pa:pye:wa	*pa:pye:-wa	AI	wait
+pa:pye:wowa	*pa:pye:w-o-wa	AI	wait
+pa:pyehe:wa	*pa:pye:-h-e:wa	TA	wait
+pa:pyehiwe:wa	*pa:pye:-h-iwe:-wa	AI	wait
+pa:šičikwaškwawa	*pa:šiči-kwa:škwaθ-wa	AI	over
+pa:šiθe:wa	*pa:šis-e:-wa	AI	past noon
+pa:šitakšawiwa	*pa:šit-a:ɣkw-eškawi-wa	AI	over
+pa:šitwo?θe:wa	*pa:šit-wohθe:-wa	AI	over
+pa:θa:pkeθamwa	*pas-a:peθk-es-amwa	TI	heat
+pa:θθeta:wa	*pa:?θi-hθeta:-wa	PT	crush
+pa:θeθowa	*pas-eso-wa	AI	hot
+pa:θhwe:wa	*pa:?θ-ahw-e:wa	TA	break

+paθikomehwe:wa	*pasi-kome:-hw-e:wa	TA	graze
+paθitiyehwe:wa	*pasi-twiye:-hw-e:wa	TA	spank
+paθšinwa	*pa:ʔθi-hšin-wa	AI	break
+paθškamwa	*pa:ʔθ-ešk-amwa	TI	break
+paka:naʔkwi	*paka:n-a:xkw-i	NA	walnut
+paka:naʔkwiwiwa	*paka:n-a:xkw-iwi-wa	AI	nut
+pakame:leme:wa	*pakam-e:lem-e:wa	TA	arrive
+pakami:te.me:wa	*pakam-i:te:-m-e:wa	TA	arrive
+pakamika:pawilotawe:wa	*pakami-ka:pawi-θotaw-e:wa	TA	arrive
+pakamika:pawiʔtawe:wa	*pakami-ka:pawi-htaw-e:wa	TA	arrive
+pakamika:pawiwa	*pakami-ka:pawi-wa	AI	arrive
+pakamipeθowa	*pakami-peso-wa	AI	arrive
+pakamipete:wi	*pakami-pete:-wi	II	arrive
+pakamiphowa	*pakami-paho-wa	AI	arrive
+pakamiptawiwa	*pakami-pahtawi-wa	AI	arrive
+pakamita:pye:wa	*pakami-ta:pye:-wa	AI	arrive
+pakamiwe:weʔšinwa	*pakami-we:we:-hšin-wa	AI	arrive
+pakapkwehamwa	*pak-apahk-we:-h-amwa	TI	knock
+pakata:nemati	*pak-ant-a:nem-atwi	II	wind
+pakatamawe:wa	*pak-ant-amaw-e:wa	TA	hit
+pakathamwa	*pak-ant-ah-amwa	TI	thresh
+pakathikani	*pak-ant-ahi-kani	NI	thresh
+pakathike:wa	*pak-ant-ahi-ke:-wa	AI	thresh
+pakathwe:wa	*pak-ant-ahw-e:wa	TA	thresh
+pakθaʔkwime:wa	*paki-hθ-a:xkwi-m-e:wa	TA	free
+pakθenamwa	*paki-hθ-en-amwa	TI	let go
+pakθene:wa	*paki-hθ-en-e:wa	TA	let go
+pakθenwi	*paki-hθen-wi	II	alight
+pakθenwi	*pa:θki-hθen-wi	II	open
+pakθeta:wa	*panki-hθeta:-wa	PT	fall
+pakθetawe:wa	*pa:θki-hθetaw-e:wa	TA	open
+pakθhamwa	*paki-hθ-ah-amwa	TI	throw
+pakθhwe:wa	*paki-hθ-ahw-e:wa	TA	throw
+pakepyeʔθa:w	*pank-epye:ʔθ-a:-w	II	rain
+pakhamawe:wa	*pak-ah-amaw-e:wa	TA	knock
+pakičiwe:penamwa	*paki-či-we:p-en-amwa	TI	throw
+pakičiwe:penamwa	*panki-či-we:p-en-amwa	TI	drop
+pakičiwe:pene:wa	*panki-či-we:p-en-e:wa	TA	drop
+pakite:leme:wa	*paki-t-e:lem-e:wa	TA	funeral
+pakitenamawe:wa	*paki-t-en-amaw-e:wa	TA	have
+pakitenamwa	*paki-t-en-amwa	TI	put

+pakitenike:wa	*paki-t-eni-ke:-wa	AI	give up
+pakithamwa	*paki-t-ah-amwa	TI	put
+pakithwe:wa	*paki-t-ahw-e:wa	TA	put
+pakito:me:wa	*paki-t-o:m-e:wa	TA	put
+pakito:tamwa	*paki-t-o:nt-amwa	TI	put
+pakkamikθe:wi	*pa:θk-axkamik-i?le:-wi	II	open
+pakkole:wa	*pa:θk-ehkweθ-e:wa	TA	open
+pakkotamwa	*pa:θk-ehkwet-amwa	TI	open
+pakkwe:liwa	*pa:θki-?kwe:-li-wa	AI	uncover
+pakšimowa	*paki-hšimo-wa	AI	set
+pakšiwela:wa	*paki-hši-wel-e:wa	TA	conduct
+pakšiweta:wa	*paki-hši-weta:-wa	PT	conduct
+pakškawiwi	*pa:θki-eškawi-wi	II	open
+pakškwa:tawe:namaw	*pa:θki-škwa:ntawe:-n-amaw-	TA	open
+pakwane:pele:wa	*pakwane:-peθ-e:wa	TA	pierce
+pakwane:peta:wa	*pakwane:-peta:-wa	PT	pierce
+pakwane:petamawe:wa	*pakwane:-pet-amaw-e:wa	TA	pierce
+pakwane:te:wi	*pakwane:-te:-wi	II	hole
+pakwane:ya?keθamwa	*pakwane:y-a:xk-es-amwa	TI	hole
+pakwanehamwa	*pakwane:-h-amwa	TI	hole
+pakwanehwe:wa	*pakwane:-hw-e:wa	TA	hole
+pakwaneškawiwa	*pakwane:-škawi-wa	AI	hole
+pakwaneškawiwi	*pakwane:-škawi-wi	II	hole
+pakwan?hike:wa	*pakwan-ahi-ke:-wa	AI	hole
+pakwhwe:wa	*pakw-ahw-e:wa	TA	mend
+pakw?hamwa	*pakw-ah-amwa	TI	mend
+pala:ča:wi	*paθ-a:č-ya:-wi	II	ruin
+pala:pame:wa	*paθ-a:pam-e:wa	TA	lose
+pala:patamawetiθowa	*paθ-a:pant-amaw-etwi-so-wa	AI	lose
+pala:patamwa	*paθ-a:pant-amwa	TI	lose
+pala:teθimikati	*paθ-a:t-esi-mikat-wi	II	ruin
+pala:teθiwa	*paθ-a:t-esi-wa	AI	ruin
+palakana:me:wa	*paθ-akana:m-e:wa	TA	miss
+pala?pile:wa	*paθ-ahpiθ-e:wa	TA	fail
+paleθhe:wa	*paθ-esi-h-e:wa	TA	miss
+paleθiwa	*paθ-esi-wa	AI	miss
+pale:leme:wa	*paθ-e:lem-e:wa	TA	miss
+pale?tawe:wa	*paθ-ehlaw-e:wa	TA	lose
+palipele:wa	*paθi-peθ-e:wa	TA	miss
+palipeta:wa	*paθi-peta:-wa	PT	miss
+pal?hamwa	*paθ-ah-amwa	TI	miss

+pal7he:wa
 +pal7hotiwaki
 +pal7škamwa
 +papa:ma:čowa
 +papa:ma:mo:wa
 +papa:ma:pye:kamowi
 +papa:ma:taka:wa
 +papa:ma:wate:wa
 +papa:mahokwiwa
 +papa:ma7te:wi
 +papa:me:lemowa
 +papa:me7tawe:wa
 +papa:mičime:wa
 +papa:mieka:wa
 +papa:mi:wa
 +papa:milešhwe:wa
 +papa:mipečiwa
 +papa:mi7θe:makanwi
 +papa:mita:čimowa
 +papa:mita:pya:θowa
 +papa:miwe:we7θenwi
 +papa:miwe7tamwa
 +papa:mo:te:wa
 +papa:mwo7θale:wa
 +pape:tanwi
 +pape:teθiwa
 +pape:tekwe:kene:wa
 +pape:tekwpaliwa
 +pape:tekwi7šinwa
 +papi:tike:ptawiwa
 +papi7θikaše:wa
 +papi7tawakočinwa
 +papi7tawakote:wi
 +papi7tawikenwi
 +papi7tawikiwa
 +papi7tawi7θenwi
 +papi7tawi7θeta:wa
 +papi7tawi7šime:wa
 +papi7tawi7šinwa
 +papi7tawipeta:wa
 +paptamawe:wa

*paθ-ah-e:wa
 *paθ-ahw-etwi-waki
 *paθ-ešk-amwa
 *papa:m-a:či-ho-wa
 *papa:m-a:mo:-wa
 *papa:m-a:pye:k-amo-wi
 *papa:m-a:taka:-wa
 *papa:m-a:wate:-wa
 *papa:m-a:hw-ekwi-wa
 *papa:m-a7te:-wi
 *papa:m-e:lemo-wa
 *papa:m-ehaw-e:wa
 *papa:mi-čime:-wa
 *papa:mi-eka:-wa
 *papa:m-i:-wa
 *papa:mi-θ-ešihw-e:wa
 *papa:mi-peči-wa
 *papa:mi-7le:-makan-wi
 *papa:mi-ta:čimo-wa
 *papa:mi-ta:pya:-so-wa
 *papa:mi-we:we:-hθen-wi
 *papa:mi-we:-ht-amwa
 *papa:m-o:te:-wa
 *papa:m-wohθal-e:wa
 *pape:t-anwi
 *pape:t-esi-wa
 *pape:tekwe:k-en-e:wa
 *pape:tekwi-pal-i-wa
 *pape:tekwi-hšin-wa
 *papi:nt-wike:-pahtawi-wa
 *papi:7si-kašye:-wa
 *papi:htaw-akočin-wa
 *papi:htaw-akote:-wi
 *papi:htawi-ken-wi
 *papi:htawi-ki-wa
 *papi:htawi-hθen-wi
 *papi:htawi-hθeta:-wa
 *papi:htawi-hšin-e:wa
 *papi:htawi-hšin-wa
 *papaθki-peta:-wa
 *pa:hpi-ht-amaw-e:wa

TA miss
 AI miss
 TI miss
 AI about
 AI about
 II about
 AI swim
 AI carry
 AI about
 II along
 AI along
 TA listen
 AI canoe
 AI spin
 AI about
 TA about
 AI about
 II about
 AI about
 AI about
 II about
 TI about
 AI crawl
 TA about
 II slow
 AI slow
 TA fold
 AI double
 AI double
 AI inside
 AI small
 AI extra layer
 II extra layer
 II extra layer
 AI extra layer
 II extra layer
 PT extra layer
 TA extra layer
 AI extra layer
 PT break
 TA laugh

+papwaškwawiwa	*papwa:-škwawi-wa	AI	fail
+pa?θenamwa	*pa:ʔθ-en-amwa	TI	crush
+pa?θene:wa	*pa:ʔθ-en-e:wa	TA	crush
+pa?θipaliwa	*pa:ʔθi-pali-wa	II	crush
+pa?θipeta:wa	*pa:ʔθi-peta:-wa	PT	crush
+pa?ka:pye:kenamwa	*pa:θk-a:pye:k-en-amwa	TI	open
+pa?ka:pye:kenike:wa	*pa:θk-a:pye:k-eni-ke:-wa	AI	open
+pa?ka:pye:kipetamwa	*pa:θk-a:pye:ki-pet-amwa	TI	open
+pa?kakana:tamwa	*pa:θk-akana:nt-amwa	TI	break
+pa?kakškwawiwa	*pa:θk-a:xkw-eškwawi-wa	AI	open
+pa?ki:kwe:wa	*pa:θk-i:nkwe:-wa	AI	open
+pa?ki:kwe?šime:wa	*pa:θk-i:nkwe:-hšim-e:wa	TA	open
+pa?ki:wa	*pa:θk-i:-wa	AI	come out
+pa?kipal?howa	*pa:θki-pali-ho-wa	AI	uncover
+pa?peθhe:wa	*pa:hp-esi-h-e:wa	TA	laugh
+pa?peθhetiwaki	*pa:hp-esi-h-etwi-waki	AI	laugh
+pa?peθime:wa	*pa:hp-esi-m-e:wa	TA	laugh
+pa?pi:wa?keθene?šinwa	*pa:hpi:w-axkesene:-hšin-wa	AI	brush
+pa?pi:wenamwa	*pa:hpi:w-en-amwa	TI	brush
+pa?pi:wene:wa	*pa:hpi:w-en-e:wa	TA	brush
+pa?pi:whwe:wa	*pa:hpi:w-ahw-e:wa	TA	brush
+pa?pi:wipele:wa	*pa:hpi:wi-peθ-e:wa	TA	brush
+pa?pi:wipeta:wa	*pa:hpi:wi-peta:-wa	PT	brush
+pa?pi:wi?θenwi	*pa:hpi:wi-hθen-wi	II	particle
+pa?pi:wi?θeta:wa	*pa:hpi:wi-hθeta:-wa	PT	brush
+pa?pi:wi?šime:wa	*pa:hpi:wi-hšim-e:wa	TA	brush
+pa?pi:wi?šinwa	*pa:hpi:wi-hšin-wa	AI	particle
+pa?pi:wiwe:penamwa	*pa:hpi:wi-we:p-en-amwa	TI	brush
+pa?pi:wiwe:pene:wa	*pa:hpi:wi-we:p-en-e:wa	TA	brush
+pa?pi:w?hamwa	*pa:hpi:w-ah-amwa	TI	brush
+pa?pilotamwa	*pa:hpi-θot-amwa	TI	laugh
+pa?pilotawe:wa	*pa:hpi-θotaw-e:wa	TA	laugh
+pa?piwinawekoθiwa	*pa:hpiwi-naw-ekw-esi-wa	AI	laugh
+pa?tawa?kwačiwa	*pa:ʔtaw-a:xkw-ači-wa	AI	stuck
+pa?tawi?šinwa	*pa:ʔtawi-hšin-wa	AI	stuck
+pa?taw?howa	*pa:ʔtaw-aho-wa	AI	evil
+pašiphike:wa	*pašipi-hi-ke:-wa	AI	stab
+pašiphwe:wa	*pašip-ahw-e:wa	TA	stab
+paškataye:paliwa	*pa:šk-ataye:-pali-wa	AI	open
+paškataye:pele:wa	*pa:šk-ataye:-peθ-e:wa	TA	open
+paškatayeškwawe:wa	*pa:šk-ataye:-škaw-e:wa	TA	open

+paškawehowa	*pa:šk-awe:-h-o-wa	AI	burst
+paškeče:θamwa	*pa:šk-ečye:-s-amwa	TI	shoot
+paškeče:θikan	*pa:šk-ečye:-s-ikan-	NI	gun
+paškeče:θwe:wa	*pa:šk-ečye:-sw-e:wa	TA	shoot
+paškeče:kana:me:wa	*pa:šk-ečye:-kana:m-e:wa	TA	crush
+paškeče:kana:tamwa	*pa:šk-ečye:-kana:nt-amwa	TI	burst
+paškeče:pele:wa	*pa:šk-ečye:-peθ-e:wa	TA	crush
+paškeče:peta:wa	*pa:šk-ečye:-peta:-wa	PT	crush
+paškečehamwa	*pa:šk-ečye:-h-amwa	TI	burst
+paškečehwe:wa	*pa:šk-ečye:-hw-e:wa	TA	burst
+paškeče?θenwi	*pa:šk-ečye:-hθen-wi	II	burst
+paškečeškawiwa	*pa:šk-ečye:-škawi-wa	AI	burst
+paškeθamawe:wa	*pa:šk-es-amaw-e:wa	TA	shoot
+paškeθoθowa	*pa:šk-esw-eso-wa	AI	shoot
+paškeθotiwaki	*pa:šk-esw-etwi-waki	AI	shoot
+paškitiye:me:wa	*pa:ški-twiye:-m-e:wa	TA	smash
+paškw?hamwa	*pa:škw-ah-amwa	TI	burst
+paškya:wi	*pa:šk-ya:-wi	II	crack
+pathoweni	*pa:ʔt-aho-weni	NI	lightning
+patkθeta:wa	*patahki-hθeta:-wa	PT	pierce
+patkečehamwa	*patahk-ečye:-h-amwa	TI	spear
+patkečeškawe:wa	*patahk-ečye:-škaw-e:wa	TA	pierce
+patkhikani	*patahk-ahi-kani	NI	pierce
+patkhotiθowa	*patahk-ahw-etwi-so-wa	AI	prick
+patkšinwa	*patahki-hšin-wa	AI	pierce
+pawa:kana	*paw-a:kan-a	NA	dream
+pawa:pkhwe:wa	*paw-a:peθk-ahw-e:wa	TA	metal
+pawawime:wa	*pawawi-m-e:wa	TA	dream
+pawawimowa	*pawawi-mo-wa	AI	dream
+pawhwe:wa	*paw-ahw-e:wa	TA	beat
+pawhwike:wa	*paw-ah-wike:-wa	AI	beat
+pawi?šinwa	*pawi-hšin-wa	AI	shake
+paw?hike:wa	*paw-ahi-ke:-wa	AI	beat
+paw?himine:wa	*paw-ahi-mine:-wa	AI	beat
+paw?škawiwa	*paw-eškawi-wa	AI	shake
+pčičike:wa	*pe?či-hči-ke:-wa	AI	mistake
+pčheθowa	*pe?či-h-eso-wa	AI	by accident
+pčhe:wa	*pe?či-h-e:wa	TA	by accident
+pčikhoθowa	*pe?č-ikahw-eso-wa	AI	by mistake
+pčipele:wa	*pe?či-peθ-e:wa	TA	by mistake
+pčipw?he:wa	*pe?či-pwi-h-e:wa	TA	poison

+pčito:ne:mowa	*pe?či-to:ne:-mo-wa	AI	by accident
+pčšinwa	*pe?či-hšin-wa	AI	by accident
+pčta:wa	*pe?či-?ta:-wa	AI	by accident
+pθelamwa	*pe?s-el-amwa	TI	by accident
+pθelawe:wa	*pe?s-elaw-e:wa	TA	by accident
+pθhamwa	*pwe?s-ah-amwa	TI	skin
+pθhwe:wa	*pwe?s-ahw-e:wa	TA	skin
+pθkani	*pwe?s-ehk-ani	NI	cloth
+pθkawe:wa	*pwe?s-ehkaw-e:wa	TA	don
+pθkawike:wa	*pwe?s-ehkawi-ke:-wa	AI	clothes
+pθkawikehe:wa	*pwe?s-ehkawi-ke:-h-e:wa	TA	clothes
+pθkawitene:wa	*pwe?s-ehkawi-t-en-e:wa	TA	don
+pθkeθenale:wa	*pwe?θ-axkesen-al-e:wa	TA	put on
+pθkeθene:wa	*pwe?θ-axkesene:-wa	AI	put on
+pe:che:wa	*pe:či-h-e:wa	TA	slowly
+pe:čičime:wa	*pe:či-čime:-wa	AI	slowly
+pe:čipeθowa	*pe:či-peso-wa	AI	slowly
+pe:čipete:wi	*pe:či-pete:-wi	II	slowly
+pe:čito:neškawiwa	*pe:či-to:ne:-škawi-wa	AI	slowly
+pe:čta:wa	*pe:či-?ta:-wa	AI	slowly
+pe:čta:wa	*pe:či-hta:-wa	PT	slowly
+pe:θkθe:wa	*pe:s-ehk-a?l-e:wa	TA	slow
+pe:θkta:wa	*pe:s-ehk-a?ta:-wa	PT	slow
+pe:khowa	*pe:nki-ho-wa	AI	clean
+pe:kipaliwi	*pe:nki-pali-wi	II	dry
+pe:kiwi	*pe:nk-i-wi	II	dry
+pe:kopye:wa	*pe:kw-epye:-wa	AI	come
+pe:kwa:wi	*pe:kw-ya:-wi	II	shallow
+pe:kwačike:wa	*pe:nkw-anči-ke:-wa	AI	dry
+pe:kwala:mwiwa	*pe:kw-aθa:mw-iwa	AI	come up
+pe:kwame:wa	*pe:nkw-am-e:wa	TA	dry
+pe:kwapiwa	*pe:nkw-api-wa	AI	dry
+pe:kwa?θe:wa	*pe:nkw-a?l-e:wa	TA	dry
+pe:kwa?ta:wa	*pe:nkw-a?ta:-wa	PT	dry
+pe:kwa?te:wi	*pe:nkw-a?te:-wi	II	dry
+pe:kwatamwa	*pe:nkw-ant-amwa	TI	dry
+pe:kwime:wa	*pe:kwi-m-e:wa	TA	wake
+pe:kwipaliwa	*pe:kwi-pali-wa	AI	wake
+pe:kwipele:wa	*pe:kwi-peθ-e:wa	TA	wake
+pe:kwitenamwa	*pe:nkwi-t-en-amwa	TI	knead
+pe:kw?he:wa	*pe:kwi-h-e:wa	TA	wake

+pe:pe:θškawiwa	*pe:pe:s-eškawi-wa	AI	go
+pe:pemočike:wa	*pe:pem-oči-ke:-wa	AI	shoot
+pe:pemotamwa	*pe:pem-ot-amwa	TI	shoot
+pe:ta:kaniwiwi	*pe:nt-a:kan-iwi-wi	II	hear
+pe:ta:koθiwa	*pe:nt-a:kw-esi-wa	AI	hear
+pe:ta:kwanwi	*pe:nt-a:kw-anwi	II	hear
+pe:takočinwa	*pe:t-akočin-wa	AI	slow
+pe:tamwa	*pe:nt-amwa	TI	hear
+pe:teθiwa	*pe:t-esi-wa	AI	slow
+pe:θečikanwi	*pesenč-ik-anwi	NI	radio
+pekiwa?keθen	*pekiw-axkesen-	NI	rubber
+pekiwe:kenwi	*pekiw-e:k-enwi	NI	rubber
+pekiw?škawite:wi	*pekiw-eškawi-te:-wi	II	dust
+pekškalati	*pek-eškaθ-atwi	II	rotten
+pekwiwa:po:w	*penkwiw-a:po:w-	NA	lye
+pekwiwa?kwi?ta:wa	*penkwiw-a:škwi-ha:-wa	PT	dust
+pekwiwe:wa	*penkwiw-e:-wa	AI	dust
+pekwiwe:wi	*penkwiw-e:-wi	II	dust
+pekwiwi:kwe?šinwa	*penkwiw-i:nkwe:-hšin-wa	AI	dust
+pele:mowa	*pele:-mo-wa	AI	turkey
+pele:wiwa	*pele:w-i-wa	AI	partridge
+pele?kawē:wa	*pele:-hkaw-e:wa	TA	turkey
+pema:čheθoweni	*pem-a:či-h-eso-weni	NI	along
+pema:čiwe:weni	*pem-a:či-h-iwe:-weni	NI	along
+pema:čhowa	*pem-a:či-ho-wa	AI	along
+pema:čhoweni	*pem-a:či-ho-weni	NI	along
+pema:θwike:wa	*pem-a:nsw-ik-e:-wa	AI	along
+pema:konaki:wa	*pem-a:konak-i:-wa	AI	walk
+pema:kone:wa	*pem-a:kone:-wa	AI	walk
+pema:mekθenwi	*pem-a:mehki-hθen-wi	II	along
+pema:mekšinwa	*pem-a:mehki-hšin-wa	AI	along
+pema:nemati	*pem-a:nem-atwi	II	along
+pema:pame:wa	*pem-a:pam-e:wa	TA	along
+pema:po:θowa	*pem-a:po:-so-wa	AI	float
+pema:pye:kamowi	*pem-a:pye:k-amo-wi	II	along
+pema:pye:wi	*pem-a:pye:-wi	II	along
+pema:pye?θenwi	*pem-a:pye:-hθen-wi	II	along
+pema:pye?šinwa	*pem-a:pye:-hšin-wa	AI	along
+pema:še:wa	*pem-a:nše:-wa	AI	go
+pema:takaškawiwa	*pem-a:taka:-škawi-wa	AI	along
+pema:teθhe:wa	*pem-a:t-esi-h-e:wa	TA	life

+pema:teθimakati	*pem-a:t-esi-makatwi	II	life
+pema:teθini	*pem-a:t-esi-ni	NI	life
+pema:teθiweniwiwi	*pem-a:t-esi-wen-iwi-wi	II	life
+pema:wale:wa	*pem-a:wale:-wa	TA	lead
+pema:wate:wa	*pem-a:wate:-wa	AI	along
+pemahoθowa	*pem-a:hw-eso-wa	AI	by canoe
+pemahwale:wa	*pem-a:hw-al-e:wa	TA	by canoe
+pemahwita:wa	*pem-a:hwi-ta:-wa	PT	by canoe
+pemakote:wi	*pem-akote:-wi	II	along
+pemapiwa	*pem-api-wa	AI	along
+pema?θama?ča:wi	*pem-a:ʔθam-ahčya:-wi	II	along
+pema?θeme:wa	*pem-a:ʔθem-e:wa	TA	along
+pema?θeta:wa	*pem-a:ʔθeta:-wa	PT	along
+pema?konamwa	*pem-a:xkw-en-amwa	TI	plan for
+pema?kone:wa	*pem-a:xkw-en-e:wa	TA	plan for
+pema?kwapiwa	*pem-a:xkw-api-wa	AI	along
+pema?kwe:wiwi	*pem-a:xkwe:-wi-wi	II	along
+pema?kwe?θe:wa	*pem-a:xkwe:-ʔle:-wa	AI	fall
+pema?kwi?θeta:wa	*pem-a:xkwi-hθeta:-wa	PT	lengthwise
+pema?kwi?šime:wa	*pem-a:xkwi-hšim-e:wa	TA	lengthwise
+pema?te:wa	*pem-a?te:-wa	II	along
+pematwe:mowa	*pem-atwe:mo-wa	AI	cry
+pemawa?kamikya:wi	*pemaw-axkamik-ya:-wi	II	land
+pemawe:wa	*pem-aw-e:wa	TA	take
+peme:lemowa	*pem-e:lemo-wa	AI	along
+peme:wa	*pem-e:wa	TA	take
+pemenamawe:wa	*pemen-amaw-e:wa	AI	care
+pemenetiθowa	*pemen-etwiso-wa	AI	care
+peme?kawile:wa	*pem-ehkawi-θ-e:wa	TA	along
+peme?kawiwa	*pem-ehkawi-wa	AI	along
+peme?košiwe:wa	*pem-ehkweš-iwe:-wa	AI	row
+peme?tawe:wa	*pem-ehtaw-e:wa	TA	along
+pemešhwe:wa	*pem-ešihw-e:wa	TA	along
+pemeštawe:wa	*pem-ešihtaw-e:wa	TA	along
+pemetone:mowa	*pem-etone:-mo-wa	AI	along
+pemetoneškawiwa	*pem-etone:-škawi-wa	AI	along
+pemhwele:wa	*pem-ahw-al-e:wa	TA	along
+pemičika:pawi:wa	*pemi-či-ka:pawi-i-wa	AI	side-to
+pemičika:pawi:wi	*pemi-či-ka:pawi-i-wi	II	side-to
+pemičipaliwa	*pemi-či-pali-wa	AI	sidewise
+pemičiptawiwa	*pemi-či-pahtawi-wa	AI	side-to

+pemiθkawiwi	*pemi-s-ehkawi-wi	Π	side-to
+pemi:paliwi	*pem-i:-pali-wi	Π	along
+pemikenwi	*pem-iken-wi	Π	grow
+pemikhe:wa	*pemi-ki-h-e:wa	TA	grow
+pemikiwa	*pemi-ki-wa	AI	grow
+pemikta:wa	*pemi-ki-hta:-wa	PT	grow
+pemile7kawē:wa	*pemi-θ-ehkaw-e:wa	TA	by
+pemile7kawike:wa	*pemi-θ-ehkaw-ik-e:wa	TA	by
+pemiššamwa	*pemi-θ-eših-amwa	TI	along
+pemiššike:wa	*pemi-θ-eših-ke:-wa	AI	along
+pemiššotiwaki	*pemi-θ-ešihw-etwi-waki	AI	along
+pemišštamwa	*pemi-θ-eših-amwa	TI	along
+pemiłotamwa	*pemi-θot-amwa	TI	accept
+pemiłotawe:wa	*pemi-θotaw-e:wa	TA	accept
+pemina:ke:wa	*pemi-na:ke:-wa	AI	by
+peminika:le:wa	*pemi-nika:θ-e:wa	TA	along
+peminika:tamwa	*pemi-nika:t-amwa	TI	along
+pemiłali7ta:wa	*pemi-pali-hta:-wa	PT	along
+pemiłaliwi	*pemi-pali-wi	Π	along
+pemiłali7he:wa	*pemi-pali-h-e:wa	TA	hither
+pemipecīwa	*pemi-pecī-wa	AI	along
+pemipecīawe:wa	*pemi-pesaw-e:wa	TA	drive
+pemipecīoweni	*pemi-peso-wen-i	NI	travel
+pemiłhekwiwa	*pemi-pah-ekwi-wa	AI	ride
+pemiłhowa	*pemi-paho-wa	AI	by
+pemiłpoćikan	*pemi-pweći-kan-	NI	plough
+pemiłpoćike:wa	*pemi-pweći-ke:-wa	AI	plough
+pemiłpole:wa	*pemi-pwel-e:wa	TA	plough
+pemiłpota:wa	*pemi-pweta:-wa	PT	plough
+pemiłpotamawe:wa	*pemi-pwet-amaw-e:wa	TA	plough
+pemiłpotawe:wa	*pemi-pwetaw-e:wa	TA	plough
+pemiłtawaw7keθenwi	*pemi-pahtaw-axkesen-wi	NI	along
+pemiłtawipale7θiwa	*pemi-pahtawi-pal-ehs-i-wa	AI	along
+pemi7θe:kan	*pemi-7le:-kan-	NA	along
+pemi7θe:makanwi	*pemi-7le:-makan-wi	Π	along
+pemi7θe:wi	*pemi-7le:-wi	Π	go round
+pemi7šime:wa	*pemi-hšim-e:wa	TA	along
+pemi7šimowa	*pemi-hšimo-wa	AI	along
+pemi7ta:wa	*pemi-hta:-wa	PT	support
+pemiłta:ćike:wa	*pemi-ta:ći-ke:-wa	AI	along
+pemiłta:ćimowa	*pemi-ta:ćimo-wa	AI	along

+pemita:pya:θowa	*pemi-ta:pya:-so-wa	AI	along
+pemita:pya:le:wa	*pemi-ta:pya:-θ-e:wa	TA	ride
+pemita:pya:lekwiwa	*pemi-ta:pya:-θ-ekwi-wa	AI	ride
+pemita:pya:šiwe:wa	*pemi-ta:pya:-š-iwe:-wa	AI	drive
+pemita:pya:tamawe:wa	*pemi-ta:pya:-t-amaw-e:wa	TA	haul
+pemita:pya:tamwa	*pemi-ta:pya:-t-amwa	TI	haul
+pemita:pye:wa	*pemi-ta:pye:-wa	AI	pull
+pemitpičikan	*pemi-t-ahpiči-kan-	NI	crossbar
+pemitpiθowa	*pemi-t-ahpiso-wa	AI	side-to
+pemitpile:wa	*pemi-t-ahpiθ-e:wa	TA	side-to
+pemitpita:wa	*pemi-t-ahpita:-wa	PT	side-to
+pemitta:wa	*pemi-t-a7ta:-wa	PT	crosswise
+pemitwo7θe:wa	*pemi-t-wohθe:-wa	AI	sidewise
+pemiwe:we:peθowa	*pemi-we:we:-pes-o-wa	AI	hear
+pemiwe:we:pete:wi	*pemi-we:we:-pete:-wi	II	heard
+pemiweta7θowa	*pemi-wet-a:7so-wa	AI	encumbered
+pemotamawe:wa	*pem-ot-amaw-e:wa	TA	shoot
+pemotiwaki	*pem-ot-wi-waki	AI	shoot
+pem7he:wa	*pemi-h-e:wa	TA	support
+pem7hetiwaki	*pemi-h-etwi-waki	AI	support
+pem7howa	*pem-aho-wa	AI	along
+pem7ška:kan	*pem-ešk-a:kan-	NI	along
+pem7škawe:wa	*pem-eškaw-e:wa	TA	along
+pemwe:keθiwa	*pem-we:k-esi-wa	AI	noise
+pemwe:tamwa	*pem-we:-t-amwa	TI	along
+pemwe:we:kati	*pem-we:we:k-atwi	II	along
+pemwe:we:keθiwa	*pem-we:we:k-esi-wa	AI	along
+pemwo7θale:wa	*pem-wohθal-e:wa	TA	along
+pemwo7θata:wa	*pem-wohθata:-wa	PT	along
+pemwo7θe:makanwi	*pem-wohθe:-makan-wi	II	along
+pena:čta:wa	*pen-a:či-hta:-wa	PT	spoil
+pena:čtamawe:wa	*pen-a:či-ht-amaw-e:wa	TA	spoil
+pena7θenwa	*pen-a:7θen-wa	II	blow
+pena7šiwa	*pen-a:7ši-wa	AI	blow
+penawe:wa	*pen-awe:-wa	AI	lay eggs
+pen7škwa:wata:wa	*pen-eskw-a:wata:-wa	PT	spread
+pen7škwe:kenamwa	*pen-eskw-e:k-en-amwa	TI	spread
+pen7škwipeta:wa	*pen-eskwi-peta:-wa	PT	unfold
+penhwe:wa	*pen-ahw-e:wa	TA	down
+penipaliwa	*peni-pali-wa	AI	downward
+penipal7howa	*peni-pali-ho-wa	AI	downward

+penipele:wa	*peni-peθ-e:wa	TA	down
+penipeta:wa	*peni-peta:-wa	PT	down
+penipočikan	*peni-pweči-kan-	NI	mill
+peni?θeta:wa	*peni-hθeta:-wa	TI	down
+peni?ta:wa	*peni-?ta:-wa	AI	down
+peniwa	*pen-i-wa	AI	bear
+peno:me:wa	*pen-o:m-e:wa	TA	down
+peno:tamwa	*pen-o:nt-amwa	TI	down
+penowa	*pen-o-wa	AI	away
+pen?hamwa	*pen-ah-amwa	TI	down
+pen?ha?kwe:wa	*pen-ah-a:ɣkwe:-wa	AI	comb
+pen?škenamawe:wa	*pen-ešk-en-amaw-e:wa	TA	down
+pen?škenamwa	*pen-ešk-en-amwa	TI	down
+pepakθenwi	*pepaki-hθen-wi	II	thin
+pepakešwe:wa	*pepak-ešw-e:wa	TA	thin
+pepakšinwa	*pepaki-hšin-wa	AI	thin
+pepakye:kati	*pepak-ye:k-atwi	II	thin
+pepikwe:ya:wi	*pepikwe:-ya:-wi	II	hollow
+pepikwiwa	*pepikw-i-wa	AI	whistle
+pepkečeθθ	*pepexk-ečye:-hs-ehs-	NA	pot-bellied
+pepkeče:wa	*pepexk-ečye:-wa	AI	pot-bellied
+pepye:θa?konamwa	*pepye:s-a:ɣkw-en-amwa	TI	pat
+pepye:θkamwa	*pepye:s-ehk-amwa	TI	wear
+pešikwa:čimwi?tawe:wa	*pešikw-a:čyemwi-htaw-e:wa	TA	straight
+pešikwa:čimw?he:wa	*pešikw-a:čyemwi-h-e:wa	TA	straight
+pešikwapiwa	*pešikw-api-wa	AI	straight
+pešikwa?kwati	*pešikw-a:ɣkw-atwi	II	straight
+pešikwe:leme:wa	*pešikw-e:lem-e:wa	TA	straight
+pešikwika:pawiwa	*pešikwi-ka:pawi-wa	AI	straight
+pešikwi?či:ke:wa	*pešikwi-hči-ke:-wa	AI	straight
+pešikwiwele:wa	*pešikwi-wel-e:wa	TA	straight
+pešikwiweta:wa	*pešikwi-weta:-wa	PT	straight
+peteke:leme:wa	*petek-e:lem-e:wa	TA	back
+peteke:letamawe:wa	*petek-e:lent-amaw-e:wa	TA	back
+petekenamwa	*petek-en-amwa	TI	back
+petekonamwa	*petekw-en-amwa	TI	fold
+petekonikani	*petekw-en-ikan-i	NI	raft
+petekwapiwa	*petekw-api-wa	AI	hunch
+petekwa?kwileče:wa	*petekw-a:ɣkwi-θenčye:-wa	AI	clench
+petekwe:kene:wa	*petekw-e:k-en-e:wa	TA	fold
+petekwikiwa	*petekw-iki-wa	AI	hump

+petekwimin	*petekwi-min-	NA	prune
+petekwipal7howa	*petekwi-pali-ho-wa	AI	double
+petekwi7kwe:wa	*petekwi-7kwe:-wa	AI	round
+petekwowa	*petekw-o-wa	AI	double
+pi:ča:wi	*pi:nč-ya:-wi	II	far
+pi:čθe:ma:wa:n	*pi:nči-7θe:ma:w-a:n-	NA	pouch
+pi:čθe:wi	*pi:nči-7le:-wi	II	inside
+pi:čikoma:kan	*pi:nči-kom-a:kan-	NA	snuff
+pi:čikome:wa	*pi:nči-kome:-wa	AI	snuff
+pi:čile7kawe:wa	*pi:nči-θ-ehkaw-e:wa	TA	inside
+pi:čiminakati	*pi:nči-minak-atwi	II	grains
+pi:čiminakeθiwa	*pi:nči-minak-esi-wa	AI	grains
+pi:čine7ke:wa	*pi:nči-neθke:-wa	AI	inside
+pi:čipeθowa	*pi:nči-pes-o-wa	AI	mud
+pi:čipele:wa	*pi:nči-peθ-e:wa	TA	inside
+pi:čiwe:penamwa	*pi:nči-we:p-en-amwa	TI	inside
+pi:čiwele:wa	*pi:nči-wel-e:wa	TA	inside
+pi:θka:kan	*pi:ns-ehk-a:kan-	NI	coat
+pi:θkawe:wa	*pi:ns-ehkaw-e:wa	TA	envelope
+pi:θkawene:wa	*pi:ns-ehkaw-en-e:wa	TA	put on
+pi:θkawe7θi	*pi:ns-ehkaw-ehs-i	NI	garment
+pi:θkawike:wa	*pi:ns-ehkawi-ke:-wa	AI	envelope
+pi:ka:kaminwi	*pi:k-a:kamy-enwi	II	grit
+pi:kala:moθowa	*pi:nk-aθa:mw-eso-wa	AI	drown
+pi:kala:mone:wa	*pi:nk-aθa:mw-en-e:wa	TA	drown
+pi:kala:mwiwa	*pi:nk-aθa:mwi-wa	AI	drown
+pi:ka7kate	*pi:nk-a:hkate:-	AI	fast
+pi:kθeta:wa	*pi:nki-hθeta:-wa	PT	spark
+pi:ke:phowa	*pi:nke:-paho-wa	AI	to weariness
+pi:kkeθamwa	*pi:kw-ehk-es-amwa	TI	break
+pi:kkole:wa	*pi:kw-ehkweθ-e:wa	TA	cut
+pi:kkotamawe:wa	*pi:kw-ehkwet-amaw-e:wa	TA	cut
+pi:kkotamwa	*pi:kw-ehkwet-amwa	TI	cut
+pi:koče:namwa	*pi:kw-ečye:-n-amwa	TI	break
+pi:koče:ne:wa	*pi:kw-ečye:-n-e:wa	TA	break
+pi:koθwe:wa	*pi:kw-esw-e:wa	TA	break
+pi:konamawe:wa	*pi:kw-en-amaw-e:wa	TA	break
+pi:košamwa	*pi:kw-eš-amwa	TI	break
+pi:košwe:wa	*pi:kw-ešw-e:wa	TA	break
+pi:kšimowa	*pi:nki-hšimo-wa	AI	to death
+pi:kškamawe:wa	*pi:kw-ešk-amaw-e:wa	TA	break

+pi:kwapiwa	*pi:kw-api-wa	AI	break
+pi:kwa?θenwi	*pi:kw-a:ʔθen-wi	II	break
+pi:kwa?keθene:paliwa	*pi:kw-axkesene:-pali-wa	AI	break
+pi:kwa?keθene:wa	*pi:kw-axkesene:-wa	AI	break
+pi:kwa?te:wi	*pi:kw-a?te:-wi	II	break
+pi:kwatamawe:wa	*pi:kw-ant-amaw-e:wa	TA	break
+pi:kwatamwa	*pi:kw-ant-amwa	TI	break
+pi:kwipaliwi	*pi:kwi-pal-i-wi	II	break
+pi:kwipoči:ke:wēni	*pi:kwi-pweči-ke:-wēni	NI	plow
+pi:kwipota:wa	*pi:kwi-pweta:-wa	PT	break
+pi:kwipota?θowa	*pi:kwi-pwet-a:ʔso-wa	AI	break
+pi:kwi?θeta:wa	*pi:kwi-hθeta:-wa	PT	break
+pi:kwitepehwe:wa	*pi:kwi-tempe:-hw-e:wa	TA	smash
+pi:kwiwe:pškamwa	*pi:kwi-we:p-ešk-amwa	TI	break
+pi:kwiwe:pškawē:wa	*pi:kwi-we:p-eškaw-e:wa	TA	break
+pi:la:kon-	*pi:l-a:kon-	XP	clean
+pi:la:piwa	*pi:l-a:pi-wa	AI	clean
+pi:lakškamwa	*pi:l-axky-ešk-amwa	TI	clean
+pi:la?či:ke:wa	*pi:l-a?či-ke:-wa	AI	clean
+pi:la?θe:wa	*pi:l-a?l-e:wa	TA	clean
+pi:la?kamikati	*pi:l-axkamik-atwi	II	clean
+pi:la?ta:wa	*pi:l-a?ta:-wa	TI	clean
+pi:leθe:letiθowa	*pi:l-es-e:lent-wiso-wa	AI	clean
+pi:le:leme:wa	*pi:l-e:lem-e:wa	TA	clean
+pi:le:letamwa	*pi:l-e:lent-amwa	TI	clean
+pi:le?θakya:wi	*pi:l-ehθak-ya:-wi	II	clean
+pi:l?howa	*pi:li-ho-wa	AI	clean
+pi:ma?kwhwe:wa	*pyi:m-a:kw-ahw-e:wa	TA	constrict
+pi:ma?kw?hamwa	*pyi:m-a:kw-ah-amwa	TI	thunder
+pi:pemwe:wa	*pi:pemwe:wa	TA	shoot
+pi:ta:kan	*pi:nt-a:kan-i	NI	pocket
+pi:ta:kone:wa	*pi:nt-a:kone:-wa	AI	inside
+pi:ta:pkwe:we:penamwa	*pi:nt-a:peθkwe:-we:p-en-amwa	TI	put
+pi:ta:wale:wa	*pi:nt-a:wale:wa	TA	haul
+pi:tahole:wa	*pi:nt-a:h-wel-e:wa	TA	inside
+pi:tawi?θenwi	*pi:ntawi-hθen-wi	II	inside
+pi:tawi?šinwa	*pi:ntawi-hšin-wa	AI	inside
+pi:teče:te:pi	*pi:nt-eče:-te:pi	AI	inside
+pi:tha:θowa	*pi:nt-ah-a:so-wa	AI	bury
+pi:tika:wale:wa	*pi:nt-wik-a:wale:wa	TA	inside
+pi:tika:waletiwaki	*pi:nt-wik-a:wale-etwi-waki	AI	enter

+pi:tika:wata:wa	*pi:nt-wik-a:wata:-wa	PT	inside
+pi:tikatawetiθowa	*pi:nt-wik-ataw-etwiso-wa	AI	inside
+pi:tikawetiwaki	*pi:nt-wik-aw-etwi-waki	AI	inside
+pi:tike:kočinwa	*pi:nt-wike:-kočin-wa	AI	inside
+pi:tike:kwaškwawa	*pi:nt-wike:-kwa:škwath-wa	AI	jump
+pi:tike:makati	*pi:nt-wike:-makat-wi	II	inside
+pi:tike:paliwa	*pi:nt-wike:-pali-wa	AI	inside
+pi:tike:pal7he:wa	*pi:nt-wike:-pali-h-e:wa	TA	inside
+pi:tike:pal7howa	*pi:nt-wike:-pali-ho-wa	AI	inside
+pi:tike:pele:wa	*pi:nt-wike:-peθ-e:wa	TA	inside
+pi:tike:pete:wi	*pi:nt-wike:-pete:-wi	II	inside
+pi:tike:ta:čimowa	*pi:nt-wike:-ta:čimo-wa	AI	inside
+pi:tike:we:penamwa	*pi:nt-wike:-we:p-en-amwa	TI	inside
+pi:tike:wo:te:wa	*pi:nt-wike:w-o:te:-wa	AI	crawl
+pi:tike:ya:mo:wa	*pi:nt-wike:y-a:mo:-wa	AI	inside
+pi:tike:ya:waθowa	*pi:nt-wike:y-a:waso-wa	AI	carry
+pi:tike:ya:wata:wa	*pi:nt-wike:y-a:wata:-wa	PT	inside
+pi:tike:ya7θenwi	*pi:nt-wike:y-a:7θen-wi	II	inside
+pi:tikehe:wa	*pi:nt-wike:-h-e:wa	TA	inside
+pi:tike7šimowa	*pi:nt-wike:-hšimo-wa	AI	into
+pi:tikeškawiwa	*pi:nt-wike:-škawi-wa	AI	into
+pi:tiko:tehotiwaki	*pi:nt-wik-o:te:-hw-etwi-waki	AI	into
+pi:tiko:tehowa	*pi:nt-wik-o:te:-ho-wa	AI	into
+pi:tkwe:pene:wa	*pi:nt-wik-we:p-en-e:wa	TA	into
+pi:tkwe:penika:θowa	*pi:nt-wik-we:p-eni-ka:so-wa	AI	into
+pi:tkwe:phamwa	*pi:nt-wik-we:p-ah-amwa	TI	into
+pi:tkwe:phwe:wa	*pi:nt-wik-we:p-ahw-e:wa	TA	into
+pi:tkwe7ta:wa	*pi:nt-exkwe:-7ta:-wa	AI	inside
+pi:to:te:wa	*pi:nt-o:te:-wa	AI	inside
+pi:wa:nemowa	*pi:w-a:nemo-wa	AI	particles
+pi:wa:pkθ	*pi:w-a:peθkw-ehs-	NI	metal
+pi:wa:pkwe:wiwa	*pi:w-a:peθkw-e:wi-wa	AI	metal
+pi:wa:pkwe:wiwi	*pi:w-a:peθkw-e:wi-wi	II	metal
+pi:wa:pkwe:ya:pi	*pi:w-a:peθkwe:-y-a:py-i	NI	metal
+pi:wa:pkwimine7θ	*pi:w-a:peθkwi-min-ehs-	NA	metal
+pi:wame:wa	*pi:w-am-e:wa	TA	scraps
+pi:wa7θenwi	*pi:w-a:7θen-wi	II	particles
+pi:wa7šiwa	*pi:w-a:7ši-wa	AI	particles
+pi:watamwa	*pi:w-ant-amwa	TI	scraps
+pi:we:leme:wa	*pi:w-e:lem-e:wa	TA	small
+pi:we:leta:koθiwa	*pi:w-e:lent-a:kw-esi-wa	AI	small

+pi:we:si:ke:wa	*pi:we:-si-ke:-wa	AI	sock
+pi:weleniwiwa	*pi:w-eleniyw-i-wa	AI	small
+pi:we?θakhikanali	*pi:w-ehθak-ahi-kan-ali	NI	shavings
+pi:we?kene?θe:wa	*pi:w-ehk-en-ehθe:-wa	AI	pick
+pi:we?kota:kanali	*pi:w-ehkwet-a:kan-ali	NI	shavings
+pi:we?kota?θowa	*pi:w-ehkwet-a:?so-wa	AI	shavings
+pi:whwe:wa	*pi:w-ahw-e:wa	TA	brush
+pi:w?hamwa	*pi:w-ah-amwa	TI	brush
+pi:w?hikwa?θowa	*pi:w-ahi-kwa:?so-wa	AI	beadwork
+pi?hikan	*pi:?s-ah-ikan-	NI	chop
+pi?θkole:wa	*pi:?s-ehkweθ-e:wa	TA	chop
+pi?θkotamwa	*pi:?s-ehkwet-amwa	TI	chop
+pi?θakana:tamwa	*pi:?s-akana:nt-amwa	TI	chop
+pi?θatamwa	*pi:?s-ant-amwa	TI	chew
+pi?θathamwa	*pi:?s-at-ah-amwa	TI	harrow
+pi?θathikan	*pi:?s-at-ah-ikan-	NI	harrow
+pi?θešamwa	*pi:?s-eš-amwa	TI	slice
+pi?θešwe:wa	*pi:?s-ešw-e:wa	TA	slice
+pi?θikahθe:wa	*pi:?s-ikah-ehθe:-wa	AI	chop
+pi?θikenwi	*pi:?s-iken-wi	II	chop
+pi?θikhamwa	*pi:?s-ikah-amwa	TI	chop
+pi?θikhe:wa	*pi:?s-ikah-e:wa	TA	chop
+pi?θikiwa	*pi:?s-ik-i-wa	AI	particles
+pi?θiminakati	*pi:?si-minak-atwi	II	pieces
+pi?θiminakeθiwa	*pi:?si-minak-esi-wa	AI	pieces
+pi?θipota?θowa	*pi:?si-pwet-a:?so-wa	AI	grind
+pi?pimetoneškawīwa	*pi:hpim-etone:-škawī-wa	AI	twitch
+pi?pimetoneškawīwi	*pi:hpim-etone:-škawī-wi	II	twitch
+pi?tawakočinwa	*pi:htaw-akočin-wa	AI	extra layer
+pi?tawakole:wa	*pi:htaw-akol-e:wa	TA	extra layer
+pi?tawakota:wa	*pi:htaw-akota:-wa	PT	extra layer
+pi?tawakote:wi	*pi:htaw-akote:-wi	II	extra layer
+pi?tawa?kamikati	*pi:htaw-axkamik-atwi	II	extra layer
+pi?tawikiwa	*pi:htawi-ki-wa	AI	extra layer
+pi?tawi?θenwi	*pi:htawi-hθen-wi	II	extra layer
+pi?tawi?θeta:wa	*pi:htawi-hθeta:-wa	TI	extra layer
+pi?tawi?θeta:wa	*pi:htawi-hθeta:-wa	PT	extra layer
+pi?tawi?šime:wa	*pi:htawi-hšim-e:wa	TA	extra layer
+pi?tawi?šinwa	*pi:htawi-hšin-wa	AI	extra layer
+pi?tawiweta:wa	*pi:htawi-weta:-wa	PT	other
+pi?tawiya:wi	*pi:htawi-ya:-wi	II	layer

+piʔtawʔʃkamwa	*pi:htaw-eʃk-amwa	TI	extra layer
+piʔtawʔʃkaw-e:wa	*pi:htaw-eʃkaw-e:wa	TA	extra layer
+piʔtawʔʃkawikani	*pi:htaw-eʃkawi-kani	NI	extra layer
+piʔtawʔʃkawike:wa	*pi:htaw-eʃkawi-ke:-wa	AI	extra layer
+piʔte:wa:pθ	*pi:ʔte:w-a:po-hs-	NI	beer
+pkame:wa	*pehk-am-e:wa	TA	burst
+pkatamwa	*pehk-ant-amwa	TI	burst
+pkaweʔta:wa	*pehk-awe:-ʔta:-wa	AI	burst
+pkθakeθiwa	*pexkw-ehθak-esi-wa	AI	hump
+pkθakya:wi	*pexkw-ehθak-ya:-wi	II	hump
+pkθhwe:wa	*paθki-hθ-ahw-e:wa	TA	remove
+pkeče:θamwa	*pehk-ečye:-s-amwa	TI	burst
+pkeče:θwe:wa	*pehk-ečye:-sw-e:wa	TA	burst
+pkečehamwa	*pehk-ečye:-h-amwa	TI	burst
+pkečehwe:wa	*pehk-ečye:-hw-e:wa	TA	burst
+pkečeʃkawiwa	*pehk-ečye:-ʃkawi-wa	AI	burst
+pkečeʃkawiwi	*pehk-ečye:-ʃkawi-wi	II	burst
+pkeθamwa	*pehk-es-amwa	TI	blow
+pkeθikan	*pehk-es-ikan-	NI	explode
+pkeθike:wa	*pehk-esi-ke:-wa	AI	explode
+pkeθowa	*pehk-eso-wa	AI	explode
+pkeθwe:wa	*pehk-esw-e:wa	TA	blow
+pkenamawe:wa	*paθk-en-amaw-e:wa	AI	remove
+pkene:wa	*paθk-en-e:wa	TA	remove
+pkeʃamawe:wa	*paθk-eʃ-amaw-e:wa	TA	remove
+pkete:wi	*pehk-ete:-wi	II	explode
+pkha:te:wi	*paθk-ah-a:te:-wi	II	remove
+pki:wa	*paθk-i:-wa	AI	remove
+pkime:wa	*paθki-m-e:wa	TA	remove
+pkipečike:wa	*paθki-peči-ke:-wa	AI	remove
+kipeta:wa	*paθki-peta:-wa	PT	remove
+pkitephwe:wa	*pehki-temp-ahw-e:wa	TA	blow
+pkočeʔθenwi	*pexkw-ečye:-hθen-wi	II	lie
+pkočeʔšinwa	*pexkw-ečye:-hšin-wa	AI	lie
+pkonamawe:wa	*paθkw-en-amaw-e:wa	TA	remove
+pkonamwa	*pehkw-en-amwa	TI	open
+pkopye:kwaʃkwalwa	*pexkw-epye:-kwa:ʃkwaθ-wa	AI	jump
+pkopye:peθowa	*pexkw-epye:-peso-wa	AI	fall
+pkopye:pete:wi	*pexkw-epye:-pete:-wi	II	fall
+pkopye:ptawiwa	*pexkw-epye:-pahtawi-wa	AI	run
+pkopye:ta:pya:le:wa	*pexkw-epye:-ta:pya:-θ-e:wa	TA	pull

+pkopye:we:penamaw	*pexkw-epye:-we:p-en-amaw-	TA	throw
+pkopye:we:penamwa	*pexkw-epye:-we:p-en-amwa	TI	throw
+pkopye:we:pene:wa	*pexkw-epye:-we:p-en-e:wa	TA	throw
+pkopye:ya:taka:wa	*pexkw-epye:-y-a:taka:-wa	AI	wade
+pkopye?θe:wa	*pexkw-epye:-?le:-wa	AI	fall
+pkopye?θe:wi	*pexkw-epye:-?le:-wi	II	fall
+pkške:wa	*paθk-ešk-e:-wa	AI	remove
+pkwa?čā:wi	*pexkw-ahčya:-wi	II	knoll
+pkwa?θenwi	*pexkw-a:θen-wi	II	together
+pkwa?kamikati	*pexkw-axkamik-atwi	NI	hill
+pkwa?kamikatonī	*pexkw-axkamik-at-weni	NI	hill
+pkwa?kamikya:wi	*pexkw-axkamik-ya:-wi	II	hump
+pkwa?kwatowe:wa	*pexkw-a:xcw-atowe:-wa	AI	ball
+pkwa?kwilečē?tamwa	*pexkw-a:xcwi-θenčye:-?t-amwa	TI	fist
+pkwa?kwilečy	*pexkw-a:xcwi-θenčy-	XP	handful
+pkwa?kwi?šinwa	*pexkw-a:xcwi-hšin-wa	AI	lump
+pkwa?pičikani	*pexkw-ahpiči-kani	NI	bundle
+pkwa?pita:wa	*pexkw-ahpita:-wa	TI	bundle
+pkwa?pitawe:wa	*pexkw-ahpitaw-e:wa	TA	bundle
+pkwa?pite:wi	*pexkw-ahpite:wi	NI	bundle
+pkwe:čike:wa	*paθkwe:-nči-ke:-wa	AI	bite
+pkwe:me:wa	*paθkwe:-m-e:wa	TA	bite
+pkwe:ne:wa	*paθkwe:-n-e:wa	TA	remove
+pkwe:nike:wa	*paθkwe:-ni-ke:-wa	AI	remove
+pkwe:šamawe:wa	*paθkwe:-š-amaw-e:wa	TA	cut
+pkwe:šamwa	*paθkwe:-š-amwa	TI	slice
+pkwe:šike:wa	*paθkwe:-ši-ke:-wa	AI	slice
+pkwe:tamwa	*paθkwe:-nt-am-wa	TI	bite
+pkwe?kole:wa	*paθkwe:-hkweθ-e:wa	TA	cut
+pkwe?kotamwa	*paθkwe:-hkwet-amwa	TI	cut
+pkwika:pawiwali	*pexkwi-ka:pawi-wali	II	stand
+pkwikanawe:wa	*pexkwi-kanawe:-wa	AI	lump
+pkwi?ta:wa	*pehkwi-hta:-wa	PT	free
+pkwitepe:wa	*pexkwi-tempe:-wa	AI	lump
+pkwiwile:wa	*paθkwi-wiθe:-wa	AI	remove
+pkw?hamwa	*pehkwi-ah-amwa	TI	open
+pkw?heθowa	*pehkwi-h-eso-wa	AI	free
+pkw?he:wa	*pehkwi-h-e:wa	TA	free
+pkw?hetiwaki	*pehkwi-h-etwi-waki	AI	free
+pkw?howa	*pehkwi-ho-wa	AI	free
+pkw?ški	*pexkw-ašk-i	NI	bushes

+po:čθe:wi	*po:nči-7le:-wi	II	enter into
+po:či:kwehwe:wa	*po:nč-i:nkwe:-hw-e:wa	TA	enter into
+po:čine7ke:liwa	*po:nči-neθke:-li-wa	AI	hole
+po:čipaliwi	*po:nči-pal-i-wi	II	hole
+po:čiwe:pene:wa	*po:nči-we:p-en-e:wa	TA	hole
+po:θčike:wa	*po:si-hči-ke:-wa	AI	load
+po:θhetiwaki	*po:si-h-etwi-waki	AI	ride
+po:θhiwe:wa	*po:si-h-iwe:-wa	TA	ride
+po:θipaliwa	*po:si-pali-wa	AI	embark
+po:θiwe:phwe:wa	*po:si-we:p-ahw-e:wa	TA	load
+po:θta7θwanwi	*po:si-ht-a:7sw-anwi	NI	load
+po:na:čimowa	*po:n-a:čyemo-wa	AI	stop
+po:na:mo:wa	*po:n-a:mo:-wa	AI	quit
+po:na:taka:wa	*po:n-a:taka:-wa	AI	swim
+po:načike:wa	*po:n-anči-ke:-wa	AI	stop
+po:nakana:me:wa	*po:n-akana:m-e:wa	TA	stop
+po:nakana:tamwa	*po:n-akana:nt-amwa	TI	stop
+po:nalo7kye:wa	*po:n-aθoxkye:-wa	AI	stop
+po:name:wa	*po:n-am-e:wa	TA	stop
+po:na7θawe:wa	*po:n-a7saw-e:wa	TA	stop
+po:na7θenamwa	*po:n-a:7θ-en-amwa	TI	stop
+po:na7θike:wa	*po:n-a:7si-ke:-wa	AI	stop
+po:na7θwe:wa	*po:n-a:7swe:-wa	AI	stop
+po:natamwa	*po:n-ant-amwa	TI	stop
+po:natwe:mowa	*po:n-atwe:mo-wa	AI	stop
+po:neθe:we7kwe:wa	*po:n-es-e:wehkwe:-wa	AI	stop
+po:nekwa:mowa	*po:n-enkwa:mo-wa	AI	stop
+po:nepye:wa	*po:n-epye:-wa	AI	cease
+po:ne7θa7θamwa	*po:n-ehθ-a:7s-amwa	TI	stop
+po:ne7kawiwi	*po:n-ehkawi-wi	II	stop
+po:niheti-waki	*po:n-i:-h-etwi-waki	AI	cease
+po:nika:pawiwa	*po:ni-ka:pawi-wa	AI	cease
+po:nike:wa	*po:ni-ke:-wa	AI	stop
+po:nina:ke:wa	*po:ni-na:k-e:-wa	AI	cease
+po:nina7ta:wa	*po:ni-na:-7ta:-wa	PT	stop
+po:nipaliwa	*po:ni-pali-wa	II	quit
+po:niptawiwa	*po:ni-pahtawi-wa	AI	stop
+po:ni7θe:wa	*po:ni-7le:-wa	AI	stop
+po:ni7θenwi	*po:ni-hθen-wi	II	quit
+po:ni7kama:wa	*po:ni-hkama:-wa	AI	stop
+po:ni7simowa	*po:ni-hsimo-wa	AI	stop

+po:ni7šinwa	*po:ni-hšin-wa	AI	quit
+po:nitehe:wa	*po:ni-te:he:-wa	AI	cease
+po:nito:neškawiwa	*po:ni-to:ne:-škawi-wa	AI	stop
+po:niwe:wa	*po:ni-we:-wa	AI	cease
+po:niwe:we:keθiwa	*po:ni-we:we:k-esi-wa	AI	cease
+po:niwe7kamwa	*po:n-iwehk-amwa	TI	stop
+po:niwe7kawe:wa	*po:n-iwehkaw-e:wa	TA	stop
+po:no:me:wa	*po:n-o:m-e:wa	TA	cease
+po:no:tamwa	*po:n-o:nt-amwa	TI	cease
+po:n7škawiwa	*po:n-eškawi-wa	AI	stop
+po:nwe:tamwa	*po:n-we:-t-amwa	TI	stop
+po:nwe:we:kati	*po:n-we:we:k-atwi	II	stop
+po:nwe:we:namwa	*po:n-we:we:-n-amwa	TI	stop
+po:nwe:we:wi	*po:n-we:we:-wi	II	stop
+po:nwe:we7θenwi	*po:n-we:we:-hθen-wi	II	stop
+po:nwo7θe:wa	*po:n-wohθe:-wa	AI	stop
+po:ta:lakenamwa	*po:nt-a:θak-en-amwa	AI	hole
+po:tawa:čikan	*po:taw-a:či-kan-	NI	chimney
+po:tawa:θowa	*po:taw-a:so-wa	AI	fire
+po:tawa:kani	*po:taw-a:kan-i	NI	fireplace
+po:tawa:te:wi	*po:taw-a:te:-wi	II	fire
+po:tawawitamwa	*po:taw-awit-amwa	TI	fire
+pokšime:wa	*po:xki-hšim-e:wa	TA	break
+pokšinwa	*po:xki-hšin-wa	AI	break
+pokškamawe:wa	*po:xkw-ešk-amaw-e:wa	TA	break
+pokškawiwa	*po:xkw-eškawi-wa	AI	break
+pokškawiwi	*po:xkw-eškawi-wi	II	break
+po7keče:šwe:wa	*po:xk-ečye:-šw-e:wa	TA	open
+po7kepye:kiwa	*po:xk-epye:ki-wa	AI	open
+po7kešwe:wa	*po:xk-ešw-e:wa	TA	open
+po7kilečehe:wa	*po:xki-θenčye:-h-e:wa	TA	break
+po7kipaliwi	*po:xki-pali-wi	II	across
+po7kitepehwe:wa	*po:xki-tempe:-hw-e:wa	TA	open
+po7koθowa	*po:xkw-eso-wa	AI	break
+po7koneθowa	*po:xkw-en-eso-wa	AI	break
+po7kwa7kamikθenwi	*po:xkw-axkamiki-hθen-wi	II	across
+po7kwa7kamikšinwa	*po:xkw-axkamiki-hšin-wa	AI	across
+po7kwa7kene:wa	*po:xkw-a:xk-en-e:wa	TA	break
+po7kwa7šiwa	*po:xkw-a:7ši-wa	AI	wind
+po7kwatelamawe:wa	*po:xkw-at-el-amaw-e:wa	TA	break
+po7kwatelawe:wa	*po:xkw-at-elaw-e:wa	TA	break

+po?kwatenamwa	*po: xkw-at-en-amwa	TI	break
+po?kwhwe:wa	*po: xkw-ahw-e:wa	TA	break
+po?kwika:te:wa	*po: xkwi-ka:te:-wa	AI	leg
+po?kwikanaweθowa	*po: xkwi-kanaw-eso-wa	AI	break
+po?kwine?ke:pele:wa	*po: xkwi-neθke:-peθ-e:wa	TA	arm
+po?kwine?ke:wa	*po: xkwi-neθke:-wa	AI	arm
+po?kwipali?ta:wa	*po: xkwi-pali-ha:-wa	PT	burst
+po?kwipal?he:wa	*po: xkwi-pali-h-e:wa	TA	burst
+po?kwi?šime:wa	*po: xkwi-hšim-e:wa	TA	break
+po?kwitepe:kana:me:wa	*po: xkwi-tempe:-kana:m-e:wa	TA	head
+po?kwiwile?šinwa	*po: xkwi-wiθe:-hšin-wa	AI	break
+po?kwowe:wa	*po: xkw-owe:-wa	AI	break
+po?pokškawiwa	*po: hpo: xkw-eškawi-wa	AI	break
+po?pokškawiwi	*po: hpo: xkw-eškawi-wi	II	break
+po?po:tawile:wa	*po: hpo:tawi-θ-e:wa	TA	blow
+po?po:tawitamwa	*po: hpo:tawi-t-amwa	TI	blow
+ppawi:wa	*pahpaw-i:-wa	AI	shake
+ppawipali?ta:wa	*pahpaw-pali-ha:-wa	PT	shake
+ppawipal?he:wa	*pahpaw-pali-h-e:wa	TA	shake
+ppawi?šime:wa	*pahpaw-hšim-e:wa	TA	knock
+ppawiwe:pene:wa	*pahpaw-we:p-en-e:wa	TA	shake
+ppaw?hamwa	*pahpaw-ah-amwa	TI	beat
+pšipaliwa	*pwe?ši-pali-wa	AI	put on
+pšipitamwa	*pwe?ši-pit-amwa	TI	put on
+pška:pa:taka:wa	*pesk-a:p-a:taka:-wa	AI	swim
+pška:pa?θenwi	*pesk-a:p-a:θen-wi	II	sail
+pška:pene:wa	*pesk-a:p-en-e:wa	TA	back
+pška:pi:wa	*pesk-a:p-i:-wa	AI	return
+pška:pi:wi	*pesk-a:p-i:-wi	II	return
+pška:pipeθowa	*pesk-a:pi-peso-wa	AI	drive
+pška:pipe:te:wi	*pesk-a:pi-pete:-wi	II	drive
+pška:piphiwe:wa	*pesk-a:pi-pah-iwe:-wa	AI	back
+pška:piptawiwa	*pesk-a:pi-pahtawi-wa	AI	run
+pška:piwele:wa	*pesk-a:pi-wel-e:wa	TA	bring
+pška:piweta:wa	*pesk-a:pi-weta:-wa	PT	bring
+pška:piwetamawe:wa	*pesk-a:pi-wet-amaw-e:wa	TA	bring
+pška:pkošiwe:wa	*pesk-a:p-ehkweš-iwe:-wa	AI	row
+pškakana:me:wa	*pesk-akana:m-e:wa	TA	bend
+pškakana:tamwa	*pesk-akana:nt-amwa	TA	bend
+pškalehe:wa	*peskale:-h-e:wa	TI	enrage
+pškale?θeta:wa	*peskale:-hθeta:-wa	PT	fire

+pškale?ta:wa	*peskale:-hta:-wa	PT	light
+pškθenwi	*peski-hθen-wi	II	bend
+pškθeta:wa	*peski-hθeta:-wa	PT	bend
+pškikwa:le:wa	*peski-kwa:θ-e:wa	TA	hem
+pškikwa:tamwa	*peski-kwa:t-amwa	TI	hem
+pškipele:wa	*peski-peθ-e:wa	TA	bend
+pškipele:wa	*peski-peta:-wa	PT	bend
+pškkawiwi	*pesk-ehkawi-wi	II	bend
+pškolamwa	*peškw-el-amwa	TI	miss
+pškonamawe:wa	*peskw-en-amaw-e:wa	TA	untie
+pškonamwa	*peskw-en-amwa	TI	untie
+pškone:wa	*peskw-en-e:wa	TA	untie
+pškošamawe:wa	*paškw-eš-amaw-e:wa	TA	cut
+pškošamwa	*paškw-eš-amwa	TI	cut
+pškošike:wa	*paškw-eši-ke:-wa	AI	cut
+pškšime:wa	*peski-hšim-e:wa	TA	curve
+pškšinwa	*peski-hšin-wa	AI	curve
+pškwa:patamwa	*peškw-a:pant-amwa	TI	see
+pškwakana:me:wa	*peškw-akana:m-e:wa	TA	miss
+pškwipele:wa	*peskwi-peθ-e:wa	TA	untie
+pškwipota:wa	*peskwi-pweta:-wa	PT	untie
+ptatene:wa	*pe?t-ant-en-e:wa	TA	by accident
+ptešamwa	*pe?t-eš-amwa	TI	by accident
+pthwiwe:wa	*pe?t-ahw-iwe:-wa	AI	by mistake
+ptškawe:wa	*pe?t-eškaw-e:wa	TA	by mistake
+ptškawike:wa	*pe?t-eškawi-ke:-wa	AI	by mistake
+pwa:lamwa	*pwa:θ-amwa	TI	fail
+pwa:le:wa	*pwa:θ-e:wa	TA	fail
+pwa:wačike:wa	*pwa:w-anci-ke:-wa	AI	fail
+pwa:wame:wa	*pwa:w-am-e:wa	TA	fail
+pwa:watamwa	*pwa:w-ant-amwa	TI	fail
+pwa:wate:wa	*pwa:w-ate:-wa	AI	carry
+pwa:wi:wa	*pwa:w-i:-wa	AI	pregnant
+pwa:wilešhwe:wa	*pwa:wi-θ-ešihw-e:wa	TA	carry
+pwa:wo:tamwa	*pwa:w-o:nt-amwa	TI	fail
+pwa:w?he:wa	*pwa:wi-h-e:wa	TA	carry
+pwapkweliwa	*pwaxp-exkwe:-li-wa	AI	through
+pwapškawiwa	*pwaxp-eškawi-wa	AI	hole
+pwapškawiwi	*pwaxp-eškawi-wi	II	hole
+pwa?pa:kone?θenwi	*pwaxp-a:kone:-hθen-wi	II	hole
+pwa?pa:kone?šinwa	*pwaxp-a:kone:-hšin-wa	AI	hole

+pwa?pa:mo:wa	*pwaxp-a:mo:-wa	AI	through
+pwa?pa:pame:wa	*pwaxp-a:pam-e:wa	TA	through
+pwa?pa:piwa	*pwaxp-a:pi-wa	AI	through
+pwa?pakana:me:wa	*pwaxp-akana:m-e:wa	TA	break
+pwa?pakana:tamwa	*pwaxp-akana:nt-amwa	TI	break
+pwa?pa?kwika:wi	*pwaxp-a:xdwi-ka:-wi	II	seep
+pwa?pe:ya:pame:wa	*pwaxpe:-y-a:pam-e:wa	TA	hole
+pwa?penamawe:wa	*pwaxp-en-amaw-e:wa	TA	break
+pwa?penamwa	*pwaxp-en-amwa	TI	break
+pwa?pene:wa	*pwaxp-en-e:wa	TA	break
+pwa?pi:wa	*pwaxp-i:-wa	AI	through
+pwa?pika:pawiwa	*pwaxpi-ka:pawi-wa	AI	through
+pwa?pipeθowa	*pwaxpi-peso-wa	AI	through
+pwa?po:te:wa	*pwaxp-o:te:-wa	AI	opening
+pwa?pwe:we?θenwi	*pwaxp-we:we:-hθen-wi	II	through
+pwa?pwe:we?šinwa	*pwaxp-we:we:-hšin-wa	AI	through
+pwe:kitwanwi	*pwe:kitw-anwi	NI	fart
+pya:pali?ta:wa	*pya:-pali-ha:-wa	PT	hither
+pya:paliwa	*pya:-pal-i-wa	AI	hither
+pya:pečiwa	*pya:-peči-wa	AI	hither
+pya:ptawiwa	*pya:-pahtaw-i-wa	AI	run
+pya:te:makati	*pya:te:-makatwi	II	arrive
+pya:te:wa	*pya:te:-wa	AI	arrive
+pya:te:wičime:wa	*pya:te:wi-čime:-wa	AI	reach
+pya:te:wipaliwi	*pya:te:wi-pali-wi	II	to the end
+pya:te:wipeθowa	*pya:te:wi-peso-wa	AI	reach
+pya:te:wipete:wi	*pya:te:wi-pete:-wi	AI	reach
+pya:te:wipokowa	*pya:te:wi-poko-wa	AI	reach
+pya:te:wiptawiwa	*pya:te:wi-pahtawi-wa	AI	reach
+pya:te:wi?ta:wa	*pya:te:wi-ha:-wa	PT	reach
+pya:te:w?he:wa	*pya:te:wi-h-e:wa	TA	reach
+pye:čiwe:wa	*pye:či-h-iwe:-wa	AI	hither
+pye:či:wa	*pye:č-i:-wa	AI	hither
+pye:čika:θowa	*pye:či-ka:so-wa	AI	forth
+pye:čiken	*pye:či-ken-	NI	vegetables
+pye:čikenwi	*pye:či-ken-wi	II	forth
+pye:čikhe:wa	*pye:či-ki-h-e:wa	TA	forth
+pye:čikiwa	*pye:či-ki-wa	AI	forth
+pye:čikta:wa	*pye:či-ki-ha:-wa	PT	forth
+pye:čile?kawē:wa	*pye:či-θ-ehkaw-e:wa	TA	hither
+pye:čime?θe:wa	*pye:či-mehθe:-wa	AI	bring

+pye:ci-myawa?θow	*pye:ci-myaw-a:ʔso-w	AI	hither
+pye:ci-myawa?te:wi	*pye:ci-myaw-a:ʔt-e:wi	II	hither
+pye:ci-myawekwati	*pye:ci-myaw-ekw-atwi	II	hither
+pye:ci-myawime:wa	*pye:ci-myawi-m-e:wa	TA	hither
+pye:ci-myawitamwa	*pye:ci-myawi-nt-amwa	TI	hither
+pye:ci-peθowa	*pye:ci-peso-wa	AI	come
+pye:ci-pete:wi	*pye:ci-pete:-wi	II	come
+pye:ci-pheti-waki	*pye:ci-pah-etwi-waki	AI	approach
+pye:ci-pkwane:piwa	*pye:ci-peθkwane:-pi-wa	AI	hither
+pye:ci-ptawiwa	*pye:ci-pahtawi-wa	AI	run
+pye:ci-ši:kane?šinwa	*pye:ci-ši:kane:-hšin-wa	AI	hither
+pye:ci-ši:kane?ta:wa	*pye:ci-ši:kane:-ʔta:-wa	AI	hither
+pye:ci-ta:ci-ke:wa	*pye:ci-ta:ci-ke:-wa	AI	hither
+pye:ci-ta:ci-me:wa	*pye:ci-ta:ci-m-e:wa	TA	hither
+pye:ci-ta:ci-ta:wa	*pye:ci-ta:cihta:-wa	PT	hither
+pye:ci-ta:pya:θowa	*pye:ci-ta:pya:-so-wa	AI	hither
+pye:ci-ta:pye:wa	*pye:ci-ta:pye:-wa	AI	hither
+pye:ci-tiye?šinwa	*pye:ci-twiye:-hšin-wa	AI	hither
+pye:ci-to:neškawiwa	*pye:ci-to:ne:-škawi-wa	AI	hither
+pye:ci-weta:wa	*pye:ci-weta:-wa	PT	hither
+pye:θa:patamwa	*pye:s-a:pant-amwa	TI	hither
+pye:θiwe:wa	*pye:s-iwe:-wa	TA	bring
+pye:θkawiwa	*pye:s-ehkawi-wa	AI	hither
+pye:letiθowa	*pye:θ-etwi-so-wa	AI	hither
+pye:lotamwa	*pye:-θot-amwa	TI	hither
+pye:lotawe:wa	*pye:-θotaw-e:wa	TA	hither
+pye:ta:θʔamika:pawiwa	*pye:t-a:θʔami-ka:pawi-wa	AI	hither
+pye:ta:kemwo?θe:wa	*pye:t-a:kem-wohθe:-wa	AI	come
+pye:ta:konaki:wa	*pye:t-a:konak-i:-wa	AI	come
+pye:ta:kone:wa	*pye:t-a:kone:-wa	AI	come
+pye:ta:po:wale:wa	*pye:t-a:po:w-al-e:wa	TA	hither
+pye:ta:pye:kamowa	*pye:t-a:pye:k-amo-wa	AI	hither
+pye:ta:pye:kiphe:wa	*pye:t-a:pye:ki-pah-e:wa	TA	hither
+pye:ta:pye?ke:wa	*pye:t-a:pye:-hke:-wa	AI	hither
+pye:ta:še:wa	*pye:t-a:nše:-wa	AI	come
+pye:ta:taka:wa	*pye:t-a:taka:-wa	AI	hither
+pye:ta:totamwa	*pye:t-a:tot-amwa	TI	hither
+pye:ta:wale:wa	*pye:t-a:w-al-e:wa	TA	hither
+pye:ta:waleti-waki	*pye:t-a:w-al-etwi-waki	AI	hither
+pye:ta:wata:wa	*pye:t-a:wata:-wa	PT	hither
+pye:ta:wate:wa	*pye:t-a:wate:-wa	AI	bring

+pye:tačike:wa	*pye:t-anci-ke:-wa	AI	hither
+pye:tahkwe:we:wa	*pye:t-ah-eθkwe:w-e:-wa	AI	bring
+pye:tahote:wi	*pye:t-a:hw-ete:-wi	II	hither
+pye:tahwiwa	*pye:t-a:hw-i-wa	AI	hither
+pye:tala:mwiwa	*pye:t-aθa:mwi-wa	AI	hither
+pye:tame:wa	*pye:t-am-e:wa	TA	hither
+pye:ta?θwe:wa	*pye:t-a:ʔswe:-wa	AI	hither
+pye:ta?kawe:wa	*pye:t-a:ʔk-aw-e:wa	TA	hither
+pye:tatamwa	*pye:t-ant-amwa	TI	hither
+pye:tatawi?tawe:wa	*pye:t-at-awi-htaw-e:wa	TA	hither
+pye:tataw?he:wa	*pye:t-at-awi-h-e:wa	TA	hither
+pye:tθemwe:wa	*pye:t-a?θemwe:-wa	AI	bring
+pye:teleshwe:w	*pye:t-eθ-ešihw-e:w	TA	hither
+pye:tenamwa	*pye:t-en-amwa	TI	hither
+pye:teša:pame:wa	*pye:t-eš-a:pam-e:wa	TA	come
+pye:teša:patamwa	*pye:t-eš-a:pant-amwa	TI	come
+pye:tešhamawe:wa	*pye:t-eših-amaw-e:wa	TA	hither
+pye:tešhamwa	*pye:t-eših-amwa	TI	hither
+pye:tešhwe:wa	*pye:t-ešihw-e:wa	TA	hither
+pye:tewe:leme:wa	*pye:tew-e:lem-e:wa	TA	arrive
+pye:tewe:letamwa	*pye:tew-e:lent-amwa	TI	arrive
+pye:thokwiwa	*pye:t-ahw-ekwi-wa	AI	hither
+pye:tkama:wa	*pye:t-ehkama:-wa	AI	smoke
+pye:tkamik	*pye:t-axkamik-	NI	hither
+pye:tkwe?šinwa	*pye:t-exkwe:-hšin-wa	AI	hither
+pye:totamwa	*pye:t-ot-amwa	TI	hither
+pye:tškošiwē:wa	*pye:t-ašk-weš-iwe:-wa	AI	bring
+pye:twe:wa:teθiwa	*pye:t-we:w-a:t-esi-wa	AI	hither
+pye:twe:we:la: tamwa	*pye:t-we:we:-θa:nt-amwa	TI	come
+pye:twe:we:mikati	*pye:t-we:we:-mikati-wi	II	hither
+pye:twe:we:pete:wi	*pye:t-we:we:-pete:-wi	II	hither
+pye:twe:we:poθowa	*pye:t-we:we:-pweso-wa	AI	hither
+pye:twe:we:wi	*pye:t-we:we:-wi	II	hither
+pye:twe:weškamwa	*pye:t-we:we:-šk-amwa	TI	hither
+pye:two?θale:wa	*pye:t-wohθal-e:wa	TA	hither
+pye:ya?šiwa	*pye:t-y-a:ʔši-wa	AI	come down
+pyephowa	*pye:h-paho-wa	AI	hither
+pyeptawiwa	*pye:h-pahtawi-wa	AI	hither
+pye?čike:wa	*pye:h-čiči-ke:-wa	AI	wait
+pye?peθowa	*pye:h-peso-wa	AI	hither
+pye?pele:wa	*pye:h-peθ-e:wa	TA	hither

+pye?petamwa	*pye:h-pet-amwa	TI	hither
+pye?pete:wi	*pye:h-pete:-wi	II	hither
+pye?pokowa	*pye:h-poko-wa	AI	hither
+pye?pokwe:makati	*pye:h-pokwe:-makat-wi	II	hither

Appendix 3

The user's Manual for PHONO is reproduced here with the permission of the author.

USING PHONO

Contents:

1. Introduction.
 - 1.1. What is PHONO?
 - 1.2. The accompanying models.
 - 1.3. Theoretical disclaimers.
 - 1.4. Distribution and feedback.
 2. Getting started.
 3. Operating an existing model.
 - 3.1. Interactive mode.
 - 3.1.1. Keyboard input.
 - 3.1.2. Screen output.
 - 3.2. Batch mode.
 4. Creating and editing models.
 - 4.1. Editing an Alphabet.
 - 4.2. Editing rule Makeup.
 - 4.2.1. Rule notation.
 - 4.2.1.1. Feature-names.
 - 4.2.1.2. Location-expressions.
 - 4.2.1.3. The seven line-types.
 - 4.2.2. Using the Makeup Editor.
 - 4.2.3. Bypassing the Makeup Editor.
 - 4.3. Editing rule Order.
 - 4.4. Editing Word Pairs (for Batch mode).
-

1. INTRODUCTION

1.1. What is PHONO?

PHONO (Version 3.2 — 1994) is a software tool (running under MS-DOS, programmed in Turbo Pascal) for developing and testing models of regular historical sound change. For this purpose, a model consists essentially of an ordered set of sound-change rules. PHONO reads the makeup of the rules from one data file, and the order of the rules from a second data file. Ancestor words are usually typed on the keyboard. PHONO applies the rules in the given order to the ancestor word and displays on the screen the successive stages of development and the final descendant form.

If you wish to test a sound-change model for which you have an ordered set of rules and a set of ancestor words, or if you teach about the operation of regular sound change, PHONO may be useful to you. The program deals with just one line of descendancy at a time (rather than deriving sister languages simultaneously). It simulates only “downstream” derivation (that is, PHONO cannot put a sound-change model “into reverse” and project upstream to generate ancestor words). Thus PHONO does not carry out directly the process of comparative reconstruction (one ancestor word from many descendants), although it may indirectly help in that enterprise by testing hypotheses.

Models may be based in any language family, and ancestor words may be either documented (as for example Latin, in the case of the Spanish model described below) or reconstructed (as for many other languages). PHONO can handle words up to 20 characters long.

1.2. The accompanying models: SPAN1 and IGPAY.

PHONO was developed in conjunction with a model for Spanish (SPAN1), which is included on the diskette. That model is derived mainly from C. P. Otero, *Evolución y revolución en romance* (Barcelona: Seix Barral, 1971), and from S. L. Hartman, “An Outline of Spanish Historical Phonology” (*Papers in Linguistics*, 7 [1974], 123-191). If you intend to alter this model, you are advised first to make a copy of it (e.g. SPAN2) and then to work with the copy. Note in particular that the files SPAN1.MAK and SPAN1.ORD both contain comments enclosed in quotation marks. These comments are invisible to PHONO, and they will be deleted if you save the model under the same name, “SPAN1”. (A future version of PHONO will enable you to edit and save comments like these, using the internal editors.) To copy the model, before starting PHONO, use the DOS “copy” command, for example as follows:

```
copy span1.* span2.*
```

Some aspects of the model SPAN1 are explained in the file
SPAN1.DOC.

The model SPANI has “Old Orthographic rules” (see Section 4.3.1 below) that enable it to receive keyboard input in standard Latin orthography (by interpreting <c> as [k], <x> as [ks], etc.). With this model, use a following colon <:> to indicate long vowels, as in “lu:na” or “vi:ta”.

Also on the diskette is a simple demonstration model, IGPAY, which receives (English) words and applies a sequence of two rules to convert the words to Pig Latin (children’s “secret” language).

1.3. Theoretical disclaimers.

Insofar as PHONO’s operation depends on assumptions about phonological theory, five disclaimers are offered:

- PHONO and models testable by it deal only with regular change; they make no claim about exceptional changes or failures to change.
- PHONO and models testable by it make no claim as to whether rules “describe”, “explain”, “cause”, “reflect”, or merely “correspond to” actual sound changes -- the relationships within and among rules remain the same.
- Except for the “Persistent” rules (see Section 4.3.3 below), PHONO treats rules as basically sequential in history -- it does not model the possible situation of rules coexisting and competing for input.
- The binary-feature notation used for encoding the makeup of rules is chosen for its precision and versatility, but there is no intention for it to represent speakers’ “psychological reality”, nor to embody any other theoretical claim.
- PHONO makes no claim about how or why sound changes are propagated through social and geographical space.

1.4. Distribution and feedback.

PHONO is intended for distribution on a monetarily “free” or cost-only basis. In exchange for its use, you are requested to do two things:

- Acknowledge the use of PHONO in any resulting publication.

-- Inform the author of any significant uses you make of the program, and of your suggestions for improving either PHONO or the model SPAN1:

Lee Hartman
Department of Foreign Languages
Southern Illinois University
Carbondale, IL 62901-4521
U.S.A.
ga5123 @ siucvmb.bitnet
ga5123 @ siucvmb.siu.edu

If you are using PHONO and you have electronic mail, you can subscribe to an electronic mailing list dedicated to discussion among users of PHONO. Send a message to the following address:

listserver @ cscs.com

Make the body (not the Subject Line) of the message as follows:

subscribe phono <your name>

Later, if you choose to cancel your subscription, send the following message to the same address:

unsubscribe phono

In using the mailing list, please note that your questions and comments to be "broadcast" to all subscribers should be sent to a different address, namely the following:

phono @ cscs.com

2. GETTING STARTED

PHONO works best when operated on a hard disk. Create a directory -- for example "PHONO" -- and use the DOS "copy" command to copy all the files from the diskette into that directory (e.g. "copy a:*. * c:\phono"). The thirteen files on the diskette are the following:

READ.ME -- This manual.

PHONO.EXE -- The PHONO program itself.

PHONO.HLP — Text for the Help screens.

SPAN1.ALF — Alphabet for the model SPAN1.

SPAN1.MAK — Rule Makeup for the model SPAN1.

SPAN1.ORD — Rule Order for the model SPAN1.

SPAN1.PAR — Word Pairs for Batch runs with the model
SPAN1.

SPAN1.DOC — Documentation of the model SPAN1.

IGPAY.ALF — Alphabet for the model IGPAY.

IGPAY.MAK — Rule Makeup for the model IGPAY.

IGPAY.ORD — Rule Order for the model IGPAY.

IGPAY.PAR — Word pairs for Batch runs with the model
IGPAY.

BETA.QES — A questionnaire for beta-testers.

To start the program, in the directory where PHONO.EXE is located, type "PHONO" and press <Enter> . After you have read the "Welcome" screen, press any key to go to the Choose Model menu. There, press the Arrow Keys to move the highlight to the name of the model you wish to work with, and then press the <Enter> key. You may choose from among the names of models or parts of models that exist on the disk, or select "(new model)" to create a new model (choosing the latter will open a field for you to type the name of your new model).

Choosing a model will take you to the Edit/Run menu for that model. In this menu, and in editing screens where initials of choices are highlighted, you may choose by pressing the corresponding letter key.

You can choose to edit any of the three parts of the model: Alphabet, Makeup, or Order. (To edit the associated Word Pairs, for Batch testing, you have to quit PHONO and use a word-processor.)

You may run the model either in the Interactive mode (the usual way — you type one word at a time on the keyboard, and PHONO shows its whole derivation on the screen), or in the Batch mode (PHONO reads a list of Word Pairs — ancestor and known descendant — from a file on the disk, and compares its derived results with known descendant words). After a run in the Batch mode, you may choose to run a Word Trace (to list the derivational history of each word) or a Rule Trace (to summarize the functional load of each rule).

Everywhere in PHONO you may quit the current procedure by pressing the <Esc> key, and

you may get Help by pressing the <F1> key. In some menus and editing screens, different Help screens are available depending on the current location of the highlight.

3. OPERATING AN EXISTING MODEL

If all three of the three essential files of a model exist – Alphabet, Makeup, and Order – then the model can be operated in the Interactive mode, which is the primary way of using PHONO. A secondary way, the Batch mode, can be run to test the integrity of a model during its development. Running in the Batch mode requires the existence of a fourth file -- <model name>.PAR – containing any number of pairs of words (ancestor and known descendant). In the Batch mode, the descendant word derived by the model is compared to the known-descendant word (input from the Pair file), and the match is recorded as either “good” or “bad”.

3.1. Interactive mode

In the Interactive mode, you type the ancestor word. Then, when you press the <Enter> key, PHONO will perform the derivation and display its successive stages. Press the Spacebar to enter another word, or <Esc> to quit.

3.1.1. Keyboard input.

The Alphabet (see Section 4.1 below) of each model provides conventions for entering ancestor words on the keyboard. By editing the Alphabet, you may assign to any keyboard character any set of feature values. In addition, you can have PHONO readjust any feature values at the beginning of each derivation, by means of “Old Orthographic” rules (Section 4.3.1). In this way, a phonological segment that has no direct equivalent on the keyboard may be represented, for example, by a digraph: the voiced bilabial fricative could be entered as the two-key sequence <h> – in conjunction with an Orthographic rule that makes /b/ fricative when followed by /h/, and then deletes the /h/. Orthographic rules exist only for the convenience of keyboard input; they have no phonological significance.

3.1.2. Screen output.

The same Alphabet used for input also is the basis of screen output, except that output is not limited to the keyboard characters, but rather can, in principle, use any of the 255 characters of ASCII code. In addition, any character of output can be modified by “feature-based diacritics”, as described in Section 4.1 below.

In the output display of the derivation, for each stage of the word, each segment is written

using the nearest equivalent character from the Alphabet's set of ASCII characters. Where there is an imperfect match between the values of the output segment and those of the character in the Alphabet, PHONO highlights the character in the screen display and follows the word with the signs and names of the features whose values differ. If more than one segment in the same string is highlighted, the respective diacritics are divided by the slashbar, "/". In order to interpret the output of a model with phonetic precision, you need to be familiar with the default feature values of that model's Alphabet.

3.2. Batch mode (with Word Trace and Rule Trace).

Each time a model is altered, it should be retested in the Batch mode — otherwise, alterations made for the sake of one set of words may cause errors for some other set. Such false steps in the development of models should be detected early, so that they can be undone.

In the Batch mode, PHONO reads — from the model's Word Pair file (e.g. SPAN1.PAR) — any number of pairs of ancestor and known-descendant words. It passes each ancestor through derivation by the model, and finally it compares the result with the known descendant from input. "Good" matches are recorded in a file GOOD.OUT, and "bad" ones in BAD.OUT.

At the beginning of Batch testing, you may choose to record each operation of each rule in a disk file (TRACE.RAW). After the Batch test, using this record, the Word Trace procedure can generate the derivational history of each word (i.e. a chronological list of the rules that act on the word); or Rule Trace can generate a summary of each rule's functional load (i.e., for each rule, a list of the words that the rule acts upon).

Unlike Word Trace, the Rule Trace procedure uses a large amount of memory. If you wish to run the model again after performing a Rule Trace, you may need to reload the model: go to the Choose Model menu and choose the model again.

If the list of Word Pairs is very long, some Batch running or Trace procedures may require that you break the list into two or more parts to be run separately.

4. CREATING AND EDITING MODELS.

When you choose to create a new model, you are first asked to type its name, with a maximum of eight characters. Then the new model will be initialized with a starting Alphabet and one "dummy" rule as a template for editing. First edit the Alphabet, then modify the dummy rule and insert additional rules. Finally, set the rules in order. Then return to the Edit/Run menu for an Interactive run.

delrel	+++--##=##++#-+++#+-----+++-++##++
strident	-----##==+-----#-----##=-++--#
voice	+++##==##++=#++#++++++++=+=++#++#
nasal	-----#-----#-----#-----#-----#
tense	+++--++-++-++-++-++-++-++-++-++-++-++
long	-----#-----#-----#-----#-----#
stress	=#-----=#-----=#-----=#-----=#-----

Initially, a new model is given a starting Alphabet, which you may edit. First edit the list of feature-names:

— To move the highlight up or down the list, press the Arrow Keys or <Page Up> or <Page Down>.

— To insert a feature-name, move the highlight to one of the existing feature-names (preferably one with values similar to those of the intended new feature), and then press <Insert>. This opens a field for you to type the name of the new feature. Finally press <Enter>. The new feature will appear below the highlighted one, and will at first duplicate its binary values. Edit the new row of value signs as described below.

— To delete a feature-name (and its row of values), move the highlight to the name and press <Delete>.

— To restore (undelete) a freshly deleted feature-name, press <Ctrl><Enter>.

— To move a feature-name up or down the list, move the highlight to the name and press <Ctrl><Page Up> or <Ctrl><Page Down>.

— To discard an entire Alphabet and restart with PHONO's internal, default Alphabet, delete all its feature-names.

After establishing the list of features, edit the character set:

— To move the highlight along the row of characters, press the Arrow Keys or <Home>, <End>, <Tab>, or <Shift><Tab>.

— To insert a new character, first locate the highlight on the existing character whose feature values are most similar to those of the new character; then press <Insert> (the values of the existing character will be duplicated in the new column). Follow the instructions that appear on the screen to choose from among the ASCII characters offered. Finally, alter any feature

values that you wish. (Normally, you will alter at least one value: a character with all the same double-signed values as one that appears earlier in the Alphabet will never be used in output.)

-- To delete a character (and its column of values), move the highlight to the character and press <Delete>.

-- To restore (undelete) a freshly deleted character, press <Ctrl><Enter>.

-- To move a character (and its column of values) to the left or right in the order of the character set, press <Ctrl> with the Left- or Right-Arrow Key.

-- To discard an entire Alphabet and restart with PHONO's internal, default Alphabet, delete all its characters.

Finally, edit the individual feature values in the body of the display. In order to choose among the four possible values (+, -, #, =), you need to understand how they are used by PHONO for input and output.

PHONO receives the input words as character strings and converts them to sets of feature values, based on the values of the Alphabet, and ignoring the single/double distinction.

For output, on the other hand, to convert the feature values back to character strings, PHONO first uses only the double signs (#/=) to assign each segment its nearest equivalent character. Then the single signs (+/-) from the Alphabet are compared with those of the output segment, and any differing values are displayed, following the character string, as "feature-based diacritics": signs and names of the differing features. For example, if the /dz/ affricate does not have a character of its own in the Alphabet's character inventory, it may appear in output as "z", highlighted -- to indicate that it differs in some way from the default /z/ -- and followed by the diacritic "-cont" to indicate that it is non-continuant, i.e. an affricate.

In editing the feature values, use keys as follows:

-- To move around the display, press the Arrow Keys or <Page Up>, <Page Down>, <Home>, <End>, <Tab>, or <Shift><Tab>.

-- To alter a value, either type the new symbol (+, -, #, =) or press the Spacebar to rotate among the four symbols.

-- To quit, when finished editing, press either <Enter> or <Esc>, and then follow the

instructions on the screen depending on whether or not you wish to save the Alphabet as edited.

4.2. Editing rule makeup.

You can alter the makeup of rules by choosing “Makeup” in the Edit/Run Menu. In order to edit rules, you need to know the basis of their notation system and how to use the Makeup Editor.

4.2.1. Rule notation.

Most phonologists are familiar with a standard rule notation based on expressions such as the following --

A -> B / C _ D

-- meaning that element A changes into element B in the environment following element C and preceding element D; in other words, each occurrence of the sequence CAD (known as the rule's “structural description”) is changed into CBD (and the change of A to B is called the “structural change”).

PHONO's rule notation portrays the structural description as a set of if-lines, and the structural change as a set of then-lines. Specifically, the format of a rule consists of four fields:

- The name line, consisting of the rule's name (with a limit of 20 characters).
- Any number (up to 25) of if-lines, labeled in order by letters (A, B, C...).
- Any number (up to 25) of then-lines, labeled in order by numerals (1, 2, 3...).
- The end line, consisting of the word “END” plus the rule name repeated.

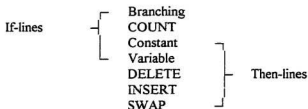
The rule in Figure 2, taken from the model SPAN1, will serve as a typical example. (The “open-mid” vowels /e/ and /ø/, portrayed here as [+low], become respectively the diphthongs /je/ and /we/.)

Figure 2: Example rule

DIPHTHONG	"Late Latin open /ε,Ω/ -> Sp. /je,we/"
A: B and C	"IF low vowel and not /a/ -"
B: +low(*)	"(here, [+low] implies vowel: /ε,Ω,a/)"
C: back(*) = round(*)	"[α back, α round] - excludes /a/"
1: -low(*) +high(*) -syllabic(*)	"THEN turn vowel to glide"
2: INSERT e (*+1)	"insert /e/ after glide"
3: stress(*+1) = stress(*)	"new /e/ gets stress of input vowel"
4: -stress(*)	"vowel-turned-glide loses stress"
END DIPHTHONG	

There are four types of if-lines and five types of then-lines — with the Constant and Variable types available for both functions — for a total of seven types, as shown in Figure 3.

Figure 3: Seven types of if- and then-lines



The COUNT-type if-line, and the DELETE-, INSERT-, and SWAP-type then-lines each begin with the name of their respective type as a keyword. If the rule has more than one if-line, these must be interrelated in a hierarchy, through the use of Branching-type lines, as described in Sections 4.2.1.3 and 4.2.2 below. In contrast, the then-lines are simply applied cumulatively and in order.

If- and then-lines are composed mainly of feature-names and location-expressions. The following sections explain these two building-blocks and how they are used to construct the

seven types of lines.

4.2.1.1. Feature-names.

Feature-names (as listed in the Alphabet) may appear in rule makeup either signed (with “+” or “-”) or unsigned. In addition to the feature-names that you arrange in editing the Alphabet, there are three abstract features – named ALL, SEGMENT, and MARK – which may be used in rules in the following ways:

-- ALL refers to all features simultaneously. It can be used in a Variable-type if-line (Section 4.2.1.3 below) to refer to identical segments (useful for finding geminates). Likewise, ALL can be used in a Variable-type then-line to carry out total assimilation (i.e. to produce geminates).

-- SEGMENT enables a COUNT-type if-line (Section 4.2.1.3 below) to count segments per se, regardless of their feature values.

-- MARK is used when the complexity of a single phonological rule requires that it be written for PHONO's interpreter as a series of two or more rules. Typically, the first of these computational rules may mark a segment [+mark] for doom, to be changed or deleted if the conditions of the following computational rule(s) are met. Generally such multi-phase rules should be followed in the Order list by a rule (named, for example, UNMARK) to return all residually marked segments to the normal status of [-mark]. This feature has nothing to do with the concept of markedness in Prague School phonology.

4.2.1.2. Location-expressions.

Location-expressions can be recognized in the text of rules partly by the fact that they are always enclosed in parentheses. There are three types of location-expressions: absolute, relative, or by feature value (the latter used only in COUNT-type lines). An absolute or relative location-expression may contain a positive or negative number, called its “magnitude”. Absolute locations in the word are indicated simply with the number of their magnitude -- positive to count from the beginning of the word, or negative to count from the end -- as in the following examples:

- (1) = initial segment of the word
- (2) = second segment
- (-3) = third-from-last segment
- (-1) = final segment

Relative location-expressions, on the other hand, are based on the fact that every application of a rule involves a scan of the word, from right to left, in search of the conditions of the rule's structural description. Each segment has its moment as the focus of the scan, during which, if the conditions are found, the change is carried out. Relative location-expressions always contain an asterisk (*) to refer to this focus segment. Additionally, the expression may contain a number for its magnitude, positive or negative, to refer to segments following or preceding the focus, as in the following examples:

(*) = the focus segment

(*+1) = the segment following the focus segment

(* -2) = the second segment to the left of the focus segment

The third type of location, by feature value, is used only in COUNT-type if-lines (see Section 4.2.1.3 below, especially the third example of a COUNT-type line, labeled "D:"). Location by feature value specifies a certain segment not by its position directly, but by the value of one of its features, as explained below.

4.2.1.3. The seven line types.

By combining feature-names with location-expressions and a few additional elements, you can form the seven types of if- and then-lines, as described below:

1) A Branching-type if-line joins two labels of following if-lines with a conjunction, either AND or OR, as in the following example:

A: B or C

Most rules have more than one if-line, and in that usual case, the first if-line, labeled "A", must be a Branching line that states the relationship between lines B and C. Then B and C, in turn, may be of other line types, or either of them may also branch, into D and E, and so on. Line A, as the top member of a hierarchy, must represent all the if-lines of the rule combined, and each if-line below line A must be represented in a Branching line somewhere above itself.

2) A COUNT-type if-line carries out a left-to-right subscan from one segment to another in the word, and it counts the number of occurrences of a given feature value between them. The COUNT-type line consists of the following six items in order:

-- the keyword COUNT

-- a signed feature-name (the target feature value)

- a location-expression for the head of the subscan
- a location-expression for the foot of the subscan
- a sign of comparison (>, <, or =)
- a numerical standard (0, 1, 2, etc.)

The example COUNT-type line below means “If the count of [+syllabic] segments from the focus segment to the word’s final segment, inclusive, is greater than two,...” That is, the rule applies only from the beginning up to, but not including, the second to last vowel in the word:

B: COUNT +syllabic (*) (-1) > 2

In a COUNT-type line, the abstract feature value [+segment] may be used to count segments per se. The following example, by insuring that the number of segments from focus to final position is one, beans, in effect, “If the focus segment is word-final...”.

In a COUNT-type line, you have the option of specifying the head or the foot of the subscan (or both) as locations by feature value. When the head of the subscan is located by feature value, it refers – from the viewpoint of the focus segment – to the nearest preceding instance of the specified value; and likewise, a location by feature value for the foot of the subscan refers to the nearest following instance. The following example, from the model SPAN1, helps to locate the post-tonic vowel of a word: it counts vowels ([+syllabic]), in a sequence reaching from the nearest stressed segment (necessarily a vowel) found to the left of the focus, to the focus itself. If the number is exactly 2 (and if the focus segment is a vowel) then the focus is on the first vowel after the stressed vowel:

D: COUNT +syllabic (+stress) (*) = 2

COUNT-type lines have a broad variety of uses, as suggested by some additional examples from the model for Spanish: to determine whether the focus segment is or is not initial or final, to identify a vowel as belonging to the penultimate syllable, to identify a word as polysyllabic or as unstressed, or to insure that no vowel intervenes between a marked agent of change and the focus segment.

3) A Constant-type if-line or then-line – the simplest line-type – is made up of one or more “units”, each consisting of a signed feature-name followed by a location-expression, as in the following example:

-nasal(1)

This expression, in an if-line, means “If the initial segment is [-nasal]...”; or, in a

then-line, "...then the initial segment becomes [-nasal]." Up to ten units can be concatenated in a Constant-type line, with the implication that they are connected by AND, as in the following example:

C: +syllabic(*) +low(*) -cons(*-1) -syllabic(*-1)

This if-line means "If the focus is a low vowel preceded by a glide..."

4) The Variable-type if-line or then-line corresponds to the Greek-letter variable signs (the so-called alpha device) introduced in N. Chomsky and M. Halle, *The Sound Pattern of English* (New York: Harper and Row, 1968), p. 83. A Variable-type line refers to the dependence of one feature value in the word on another, as either equal or opposite. Used as an if-line, the Variable line refers to various conditions of interdependency among feature values. As a then-line, it appears in many rules of assimilation (signs equal) or dissimilation (signs opposite). The Variable-type line consists of five elements, as follows:

- an unsigned feature-name (the focus feature)
- a location-expression (the focus location)
- a sign of equality (=) or inequality (≠)
- a second feature-name (the standard feature)
- a second location-expression (the standard location)

The following example, repeated from the rule DIPHTHONG (Section 4.2.1 above) is a Variable-type if-line specifying the natural class of the back-rounded and the front-unrounded vowels together – [α back, α round]:

C: back(*) = round(*)

Line 3 of the same rule is also of the Variable type, a then-line, stating that the value of stress on the segment following the focus segment is made the same as the value of stress on the focus segment (the effect of this, together with line 4 to unstress the focus segment, is to shift stress one segment to the right).

3: stress(*+1) = stress(*)

A then-line for dissimilation uses the sign of inequality (≠). The following example line expresses dissimilation of voicing from the following segment:

1: voice(*) ≠ voice(*+1)

5) A DELETE-type then-line deletes a specified segment. It consists of the keyword DELETE and a location-expression. The following example would delete the word-final

segment:

2: DELETE(-1)

6) An INSERT-type then-line inserts a character from the Alphabet in a given location in the word. It consists of three elements:

- the keyword INSERT
- the character to be inserted (the “new segment”)
- a location-expression

The following example – repeated here from the rule DIPHTHONG (Section 4.2.1 above) – inserts the segment /e/ immediately after the focus segment:

2: INSERT e (*+1)

In the event that the desired new segment differs in some feature values from the character inserted from the Alphabet, these values can be adjusted by means of additional then-lines – of the Constant or Variable types – in the same rule.

In INSERT-type lines, the location-expression referring to the focus segment alone (*) means the focus segment is replaced by the new segment. Otherwise, expressions with the focus plus or minus a number – (*+2), (*-1), etc. – mean insertion at the indicated segment boundary, counting right or left of focus segment respectively.

7) A SWAP-type then-line interchanges the positions of two segments in the word (metathesis). Its format consists of three elements: the keyword SWAP and the location-expressions of the two segments to be interchanged. The following example line interchanges the focus segment with the segment to its left:

4: SWAP (*) (*-1)

4.2.2. Using the Makeup Editor.

To compose a rule in PHONO's notation, first think of its structural description (i.e. the segment to change and the environmental conditions necessary for the change) as one great if-clause, to be represented by the first if-line, labeled “A”. Then subdivide those conditions successively by means of Branching-type if-lines until you reach conditions that are simple enough to be represented by Constant-, Variable-, or COUNT-type if-lines. These bifurcating divisions may be between the “triggers” and the “target” of the change (or, in

non-ballistic terms, the “agents” of change and the “mutand” — that which is to be changed), between the left and right sides of the environment, between consonantal and vocalic aspects of the environment, between front and back vowels as the focus of change, or between any other pair of complementary conditions. See the rules in the models SPAN1 and IGPAY for examples.

To use the Makeup Editor, in the Edit/Run Menu, move the highlight into the Edit portion of the menu and choose Makeup. First you will see the list of rule names in alphabetical order: use the Arrow Keys, etc., to move the highlight up or down the list to choose the rule that you will edit, then press <Enter> to open that rule for editing. With the rule's text on screen, use the Arrow Keys, <Page Up>, <Page Down>, <End>, and <Home> to move the highlight around to any part of the rule that you wish to alter.

-- Name: To rename the rule, locate the highlight on the name line, then press <Tab> to open a field for retyping the name. In naming rules, use a single word or use the underscore () between words (“FINALSTRESS” or “FINAL_STRESS”, not “FINAL STRESS”).

-- Line-type: To change the type of a line — among the three non-Branching types of if-lines or the five possible types of then-lines — locate the highlight on either the label (A, B, C..., 1, 2, 3...) or the keyword, if there is one (COUNT, INSERT, DELETE, SWAP); then press <Tab> to rotate among the line-types. (Caution: Each new line type is given arbitrary values, to be edited — so returning to the original line type does not guarantee a return to its original values.)

-- Feature-name: With the highlight located on a feature-name, press the Spacebar to advance one step along the list of feature-names in the Alphabet; or press <Tab> to open a menu showing all the feature-names to choose from.

-- Location-type: To change the type of a location-expression — for example, to rotate from absolute “(-1)” to relative “(*-1)”, or to by-feature-value “(-syllabic)” — position the highlight on the location-expression, and then press <Tab>.

-- Location-magnitude: To change the magnitude of a location-expression (rotating among values from -3 to 3), position the highlight on the location-expression, and then press the Spacebar.

-- New segment: In an INSERT-type then-line, with the highlight on the character to be inserted, press the Spacebar or <Back Space> to move forward or back along the line of characters in the Alphabet; press <Tab> to see the character set of the Alphabet as a menu to choose from; or simply press the corresponding letter key.

– Signs, conjunctions, numerals: On the remaining items, the Spacebar and <Tab> have the same effect, namely toggling or rotating among the possible values: = and \=, and/or, < = >, 0/1/2/3, +/-.

– Inserting lines: With the highlight located on an if- or then-line's label (A, B..., 1, 2...), press the <Insert> key to insert a new line. The rule will gain a line of the Constant type, consisting of one unit – a signed feature-name and a location-expression – e.g. +syllabic(*); then you can alter the line-type or any of its contents as described above. If the highlight is on the label of a then-line (1, 2, 3...), the new line will be inserted in numerical order after it. But if the highlight is on the label of an if-line (A, B, C...), then that line is changed to a Branching-type line, joining two new lines: one with the contents of the former highlighted line, and the other being initialized as a Constant-type line, with arbitrary contents open to alteration as described above.

– Deleting lines: With the highlight located on an if- or then-line's label, press the <Delete> key to delete that line. When an if-line is deleted, its “sibling” line moves up the hierarchy to replace the Branching-type contents of their mutual “parent” line. A Branching-type if-line cannot be deleted except by deleting both of its daughter lines.

– Inserting Constant-type units: In a Constant-type line, to insert additional units, (1) move the highlight to one of the existing units in the line (i.e., to any one of the unit's three parts), and (2) press the <Insert> key. PHONO inserts a new unit after the one with the highlight, and initializes it with arbitrary contents – e.g. “+syllabic(*)”. You may then alter any of its parts in the usual way.

– Deleting Constant-type units: In a Constant-type line, to delete one of the units (if there are more than one), (1) move the highlight to any part of the unwanted unit, and (2) press the <Delete> key.

– Changing order of lines: To change the order of if- or then-lines, locate the highlight on the label of the line to move, then press <Ctrl><Page Up> or <Ctrl><Page Down> to move the line up or down in the rule. In the if-section of the rule, such changes are for esthetic purposes only: they do not affect the logic of the rule. But changing the order of then-lines reorders their application, and thus may change the functioning of the rule.

– Exiting the rule: When you have finished editing the rule, to accept the editing changes you have made and incorporate the rule into the current model in memory, press <Enter>. Conversely, if you want PHONO to forget your editing and leave the rule as it was, press <Esc>. Either of these keys will return you to the Makeup Editor's alphabetized list of rule names.

-- Exiting the Editor : When you have finished editing all the rules that you want to, and have returned to the list of rule names, you may save the new version of the model to the disk by pressing <Alt><S>. With or without saving, leave the Makeup Editor by pressing <Esc>.

4.2.3. Bypassing the Makeup Editor.

If you have trouble with the Makeup Editor, or if for any other reason you prefer to edit the Makeup of rules "directly" with a word-processor, that option is available. Set the processor to produce a simple ASCII text file, with no formatting characters. The appearance of rules in Makeup files (e.g. SPAN1.MAK) is essentially the same as in the Makeup Editor. If you try editing rules "freehand" in this way, there are a few Error Messages available to help you find typographical errors.

4.3. Editing rule Order.

The Order Editor displays two lists: the chronological Order list on the left, and the alphabetized Supply list on the right. A large arrow in the middle points to the currently active list, and you can toggle it with the Arrow Keys or the <Tab> key. You can add a rule name to the Order list by choosing it from the Supply list, copying it into the Buffer (at the top of the screen), and finally installing it in the Order list.

The Order list determines how the model will be operated: which rules will be applied, and in what order. Every rule name in the Order list must come from the Supply list -- but any rule from Supply may be used in Order once, more than once, or not at all.

The Order list is structured in three labeled sections, corresponding to the three major types of rules that PHONO recognizes: (1) Old Orthographic, (2) Phonological, and (3) New Orthographic. The Phonological rules belong to the sound-change model proper, while both kinds of Orthographic rules merely serve to interpret character input -- from the keyboard or from the Pair file.

Old Orthographic rules serve to adjust the feature values of ancestor word input, both in the Interactive and in the Batch mode.

New Orthographic rules, on the other hand, are used only in the Batch mode. They likewise have an interpretive function, with regard to the spelling of the known-descendant words in the pairs tested.

In standard Latin orthography, for example, the letter C is thought to have corresponded to a velar [k] during the early history of that language. But in PHONO's Standard Alphabet,

the unmodified /c/ is a palatal affricate. Nevertheless, Latin can be input in its own orthography by using a series of Old Orthographic rules that includes a rule such as CEE_KAY, to convert the Alphabet's voiceless palatal affricate (as it is first misinterpreted) to the desired velar stop.

Meanwhile, the same letter C — before E or I — in some dialects of modern Spanish stands for the interdental fricative [θ]. Starting with the same palatal affricate from the Alphabet, a New Orthographic rule bridges the gap, making it possible in the word pairs of SPAN1.PAR to spell the known Spanish descendant word in standard orthography.

Sandwiched between the two kinds of Orthographic rules are the Phonological rules — that is, the rules that form the significant body of the sound-change model. Such rules are normally considered transient (acting no more than once in a derivation), but you can mark any of them persistent (repeating their changes throughout the derivation whenever their conditions occur) according to the needs of your model. For example, in the model SPAN1, the rule HOMORGANIC (i.e. syllable-final nasal assimilation) is marked as Persistent. Accordingly, nasal assimilation is carried out initially on the input Latin orthography, as well as centuries later in consonant clusters brought together by vowel-deletion. (The term persistent is defined by Wallace Chafe — IJAL 34 [1968], 131 — and is used here in its diachronic sense.)

To install a rule from the Supply list into the Order list —

- press <Tab> or the Left or Right Arrow keys to toggle the large arrow between the two lists and choose the Supply list, on the right;

- in the Supply list, move the highlight to the name of the rule that you want to install in Order;

- press <Enter> to copy that name into the Buffer;

- toggle the large arrow to the Order list, on the left;

- move the highlight to just above the desired position in Order, and press the <Insert> key.

To delete a name from the Order list —

- move the highlight to the name;

- press the <Delete> key.

To move a name up or down the Order list —

- move the highlight to the name;
- press <Ctrl><Page Up> or <Ctrl><Page Down>.

To mark or unmark a rule as Persistent (see above) –

- in the “Phonological” part of the Order list, move the highlight to the name;
- Press <P>.

4.4. Editing Word Pairs (for Batch Mode).

Word Pairs (ancestor and known descendant, for testing in the Batch mode) are held in a file (e.g. SPAN1.PAR or IGPAY.PAR). In the present version of PHONO, you may look at the Word Pairs without quitting the program; but to edit them you must exit PHONO and use a word-processor.

To look at the word pairs:

- go to the Edit/Run Menu;
- locate the highlight on Pairs in the Edit section of the menu;
- press <L>.

To edit the word pairs:

- exit PHONO;
- using a word-processor set to produce an unformatted ASCII text file, open the Pair file (SPAN1.PAR, IGPAY.PAR, etc.);
- edit the word pairs as you wish, making sure each pair is complete, in ancestor/descendant order, and making sure to use only the orthographies that you have arranged for the model to recognize through its Alphabet and its Old and New Orthographic rules, if any.

NOTE: When using your word-processor to create the Word Pair file, after writing the last item, press the Spacebar or <Enter>; this is to insure that PHONO can read the last word entirely.



