COLORADO TECHNICAL UNIVERSITY

ENABLING A LEGACY MORPHOLOGICAL PARSER
TO USE DATR-BASED LEXICONS

VOLUME 2 of 2

APPENDICES

A DISSERTATION SUBMITTED TO
THE GRADUATE COUNCIL
IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

BY

MICHAEL A. COLBURN

M.C.I.S., University of Denver, Denver, CO, 1991
M.A., University of Texas, Arlington, TX, 1981
B.A., Rockmont College, Denver, CO, 1975

DENVER, COLORADO
DECEMBER 1999

ABSTRACT

These appendices contain foundational work conducted for the author's research. The research is an investigation of the feasibility and desirability of using DATR with AMPLE, a legacy morphology exploration tool developed by the Summer Institute of Linguistics. The research demonstrates the feasibility of using DATR with AMPLE by showing how DATR can be used to encode the lexical information required by AMPLE and by presenting an interface between AMPLE and DATR-based lexicons that does not require modification of AMPLE itself. The desirability of using DATR with AMPLE is shown by demonstrating that there are generalizations that cannot be captured using AMPLE’s lexical knowledge representation language (LKRL) that can be captured in DATR, thereby reducing redundancy of lexical information.

Appendix 1 is an overview of Ogea morphophonemics and morphotactics. Appendix 2 is an Ogea AMPLE lexicon that supports the parsing of all Ogea examples found in Appendix 1. Appendix 3 is the DATR version of the Ogea lexicon. Appendix 4 is both an AMPLE and DATR version of a subset of a Yalálag Zapotec AMPLE lexicon. Finally, Appendix 5 lists various Perl scripts used for the research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ABBREVIATIONS

The following abbreviations are used to gloss language examples.

1	First person
2	Second person
3	Third person
B	Benefactive
cert	Certainty
Contra	Contrafactive
d	Dual
Hab	Habitual
imp	Imperative
int	Intensive
inf	Infinitive
Nom	Nominative
O	Object
p	Plural (note: a number after the p indicates the 1st or 2nd type of plural)
P	Possessive
perf	Perfective
pred	Predictive
s	Singular
S	Subject
Sj	Subjunctive
SR	Switch Referent
Tf	Future Tense
Tip	Immediate Past Tense
Tp	Present Tense
Trp	Remote Past Tense
TO	Temporal overlap
TS	Temporal Succession
vAux	Verbal Auxilliary
vAuxR	Verbal Auxilliary Root

APPENDIX ONE

AN OVERVIEW OF

OGEA MORPHOPHONEMICS AND MORPHOTACTICS

1.0 Description

The purpose of this appendix is to provide information about Ogea morphophonemics and morphotactics. Information is provided at a level of detail sufficient for the reader to follow the discussion about the Ogea AMPLE-based lexicon and the Ogea DATR-based lexicon found in Appendices Three and Four. An additional purpose of this appendix is to provide a description of the language's morphophonemics and morphotactics since little has been published on the Ogea language.

Ogea[1] is a Papuan (non-Austronesian) language spoken by approximately 700 people living in an area 18 kilometers south of the town of Madang, in the Madang Province of Papua New Guinea. The Ogea live in four villages—Garima, Dogia, Balama, and Erima, bounded by the Golgol and Yawor rivers, and the Astrolabe Bay. Ethnically, the Ogea are Melanesians. There is evidence that Polynesians migrated through New Guinea in pre-European times. Although not an Austronesian language, Ogea contains a number of Austronesian cognates—perhaps loan words—indicating the possibility of interaction between the Ogea and the migrating Polynesians. The first recorded contact with the Ogea by a European was likely made by the Russian scientist, Miloucho-Maclay, who describes visits to several Ogea villages between 1871 and 1883 in his diary [5]. Ogea has at least one Russian loan-word, *sapora* 'axe, hatchet', evidence perhaps that Miklouche-Maclay introduced metal to the Ogea. In pre-European times, the Ogea were a stone-age society.

Linguistically, Ogea is classified as follows [3]:

Trans-New Guinea Phylum
Madang-Adelbert Range
Madang
Rai Coast
Nuru Family
Ogea

Ogea linguistic data were collected by this author while living among the Ogea between 1977 and 1986 under the auspices of the Summer Institute of Linguistics, and during a visit in 1999. The original grammatical analysis was presented by this author in [1] and was based on a collection of 40 pages of native authored text. The original analysis was reviewed by Dr. Ger Reesink, then of the Summer Institute of Linguistics in Papua New Guinea, and now of the Leiden University in the Netherlands. For purposes of this dissertation, this author did a reanalysis, using a collection of native-translated[2] texts containing more than 223,000 words. Where paradigms could not be completed using texts, data were elicited from native speakers. The original analysis was based on the minority dialect of Ogea (spoken in Erima village). The analysis presented here is based on the majority dialect (spoken in Garima, Dogia, and Balama). Deviations from

[1] Also known as Erima in the literature. However, speakers of the language refer to it as Ogea, not Erima (which is a village name).

[2] The use of translated texts is far from ideal, but few native-authored texts are available. The texts used were translated by a native Ogea speaker. The use of such texts for morphological studies is likely more defensible than using them for studies of syntax or discourse. Influence of the source language is less likely to impact morphology than, say, syntax or discourse. To date no morphological contradictions have been found between the large body of translated texts available to this author and the small body of native authored texts that are available. Also, all examples used in this appendix have been verified by a native Ogea speaker.

the original analysis will be noted. To the best of this author's knowledge, no analysis of Ogea phonology, grammar, or morphotactics exists apart from that of this author.

Phonemically, Ogea has a 15 vowel system with 17 consonants. Syntactically, Ogea is a Subject-Object-Verb (SOV) language, with adjectives following nouns, and deictics following adjectives—the reverse of English. Morphologically, Ogea is a highly inflected, suffixing language, with most of the complexity occurring with verbs. There are over 100 basic verbal suffixes, the number of which is significantly multiplied by allomorphic variants. Ogea sentences are often composed of chains of verbs, with suffixes indicating sentence *medial* versus *final* positions. Ogea verbs encode inter-clausal temporality (temporal succession—one action occurs following another—and temporal overlap—actions occur simultaneously). They also encode switch reference. Switch reference indicates whether the referents of the clause in question are referents in the following clause. It is useful to classify Ogea verbal suffixes into two major categories: endocentric and exocentric, following the lead of Staalesen [6] and Wells [7]. Endocentric suffixes occur between the verb root and the exocenter. Endocentric suffixes include manner, object, and benefactive suffixes, among others. The same set of endocentric suffixes are used with varying sets of exocentric suffixes. The endocenter itself is composed of the verb root plus the endocentric suffixes. Exocentric suffixes encode inter-clausal temporality, tense, mood, subject, and switch reference. They are termed exocentric because they may contain suffixes that relate to the clause that follows (i.e., inter-clausal temporality and switch reference relate the current clause to the one that follows it).

In the sections below, Ogea morphophonemics will be presented first, followed by an overview of Ogea morphotactics. Note that in the examples provided throughout Appendix One, when multiple glosses are possible, sometimes only one is selected in order to allow the example to fit on a single line. Appendix Three or Four may be consulted to see the full range of possible glosses for any particular morpheme.

1.1 Morphophonemics

The original analysis of Ogea made by this author did not include morphophonemic rules. Based on the current analysis, there are four morphophonemic rules in Ogea. The rules are as follows:

- Rule 1. Syncope
- Rule 2. Glide Formation
- Rule 3. Epenthetic Consonant Insertion
- Rule 4. Nasal Assimilation

The rules are based on the orthographic representation of Ogea, which also happens to be phonemic (see Table 1). In this scheme, there is partial ordering of the rules in that Rule 2 must apply before Rules 3 or 4. The classes used for Ogea morphophonemic rules are shown in Table 2. Note that the verb roots listed for Morpheme Class One are a closed class, whereas the verb roots in Morpheme Class Two are an open class. In the statement of morphophonemic rules, the '+' symbol is used below to indicate a morpheme boundary, and a 'Ø' indicates the loss of a segment.

Table 1 Ogea Phonemes and Orthography

Type		Values	
Vowels	Normal	Phonemic	a e i o u
		Orthographic	a e i o u
	Nasalized	Phonemic	ã ẽ ĩ õ ũ
		Orthographic	a e i o u
	Lengthened	Phonemic	a: e: i: o: u:
		Orthographic	aa ee ii oo uu
Consonants	Nasals	Phonemic	m n ɲ ŋ
		Orthographic	m n ny ng
	Plosives	Phonemic	t k b d g
		Orthographic	b t k d g
	Fricatives	Phonemic	ɸ s β
		Orthographic	f s w
	Affricates	Phonemic	dʒ
		Orthographic	j
	Trill	Phonemic	r
		Orthographic	r
	Approximants	Phonemic	w l y
		Orthographic	w l y

Table 2 Classes Used for Ogea Morphophonemic Rules

Class	Description		
A	Alveolar plosives		
B	Bilabial plosives		
C	All consonants		
N	Nasals		
V	All vowels		
X	Velar plosives		
MC1	Verb Roots	ati-	make
		aule-	carry on the shoulder
		awa-	leave
		bisi-	bathe
		biti-	burn
		dage-	tie
		heige-	appear, arrive
		huge-	wash
		inyi-	be, remain
		isi-	hear
		mani-	go up
		mere-	cry
		mini-	go down
		nyi-	eat
		te-	get
		tafa-	put
		tani-	place inside
		ume-	die
		uye-	dance
		yafa-	sit
		Table 23	see
		Table 24	hit
		Table 25	get (a person)
		Table 26	give
	Polysyllabic Object and Benefactive Suffixes	-hagi-	B1p
		-hani-	B2s
		-hari-	B1d
		-higi-	O1p
		-hini-	O2s
		-hiri-	O1d
		-nagi-	B3p
		-nari-	B3d
		-nigi-	O3s
		-niri-	O3d
		-tagi-	B2p
		-tari-	B2d
		-tigi-	O2p
		-tiri-	O2d
MC2	Verb Roots	Those not listed in MC1	
	Suffixes	Those not listed in MC1	

1.2.1 Rule 1. Syncope

This rule states that the final vowel of a morpheme is dropped when the morpheme precedes another morpheme that begins with a vowel, and is otherwise retained. The rule only applies to morphemes of Morpheme Class 1 (see Table 2).

(1) $V]^{MC1} \rightarrow \emptyset \ / \ \_ \ +V$

Some examples of Rule 1 are shown below. The examples are given in pairs. The first example of each pair shows the loss of the vowel, and the second example of each pair shows the retention of the vowel. For each example, the morpheme in bold is the one to which the rule applies.

(2) a. **yafa**- + -o- + -na → yafona
sit -Ind.Tf -Ind.S3s
'he/she will sit'

b. **yafa**- + -de → yafade
sit -Ind.S3p
'he/she sits'

(3) a. **te**[3]- + -a- + -i → tai
get -Ind.Trp -Ind.S3s
'he/she got'

b. **te**- + -ni → teni
get -Ind.S1/2s
'I get'

(4) a. le- + -**hini**- + -e- + -ye → lehineye
speak- -O2s -Ind.Trp -Ind.S1/2p
'We spoke to you'

b. le- + -**hini**- + -ni → lehinini
speak- -O2s -Ind.S1/2s
'I speak to you'

[3] A variant of /te-/ is /tee-/. This appears to be a stylistic lengthening of the vowel.

1.2.2 Rule 2. Glide Formation

This rule states that in the case of verb roots and suffixes whose shape is CVi, the final /i/ becomes /y/ preceding a suffix that begins with a vowel, and remains /i/ otherwise.

(5) i → y / +CV_ +V

Examples of Rule 2 are provided below. As with Rule 1, the examples are given in pairs.

(6) a. te- + -**hai**- + -o- + -na → tehayona
get -B1s- -Ind.Tf- -Ind.S3s
'He/she will get (it) for me'

b. te- + -**hai**- + -g- + -ou → tehaigou
get- -B1s- -Hab- -Nom
'One who get (things) for me'

(7) a. **goi**- + -o- + -na → goyona
go- -Ind.Tf- -Ind.S3s
'He/she will go'

b. **goi**- + -g- -a- + -I → goigai
go- -Hab- -Ind.Trp -Ind.S3s
'He/she habitually went'

(8) a. **jai**- + -e- + -ye → jayeye
get.up- -Ind.Trp- -Ind.S1/2p
'We/you (plural) got up'

b. **jai**- + -du → jaidu
get.up- -TS
'Having gotten up'

Note that Rule 2 does not apply to polysyllabic morphemes that end with /Vi/. Instead, Rule 3 applies.

1.2.3 Rule 3. Epenthetic Consonant Insertion

This rule states that a /w/ is inserted at the beginning of morphemes belonging to Order Class 8[4] if they begin with a vowel and if they follow a morpheme of Morpheme Class 2 that ends in a vowel. Otherwise, a /w/ is not inserted.

(9) $V]^{OC8} \rightarrow wV\ /\ V+]^{MC2}\ \_$

Some examples of Rule 3 are:

```
(10)  a. oofa-      + -ou                  → oofawou
         watch.over-  -Nom
         'One who watches over'

      b. oofa-     + -g-   + -ou           → oofagou
         watch.over-  -Hab    -Nom
         'One who habitually watches over'

(11)  a. ilei- +     -a- +     -i          → ileiwai
         press.down- -Ind.Trp- -Ind.S3s
         'He/she pressed (it) down'

      b. ilei- +     -nigi- + -a- +     -i  → ileinigai
         press.down- -O3p     -Ind.Trp  -Ind.S3s
         'He/she pressed (it) down'

(12)  a. yafa- + -de- + -e- +      -ge      → yafadewege
         sit-    -well  -Contra.S1s-Contra
         'If I had sat properly…'

      b. yafa- + -e- +      -ge            → yafege
         sit-    -Contra.S1s-Contra
         'If I had sat…'
```

1.2.4 Rule 4. Nasal Assimilation

With few exceptions, Ogea syllables are open. That is, they do not end with a consonant. One exception[5] occurs with certain verb roots that end with a nasal. When such roots combine with a suffix that begins with a stop, the nasal assimilates the point of

[4] See Table 5 through Table 9. An order class 8 suffix is the first suffix of the endocenter, if the endocenter does not have a temporality suffix. Otherwise it is the second suffix of the endocenter.

[5] The other exception is certain subject suffixes in the future tense when they precede the auxiliary verb *bo-* 'intent'. This is discussed in section 1.2.3.5.6 Future Tense Suffixes.

articulation of the stop that begins the morpheme that follows. If the root combines with a suffix that begins with any other consonant, the vowel preceding the nasal becomes nasalized, and the nasal segment is deleted.

```
(13)  VN → Vm  / _ -[B]
         → Vn  / _ -[A]
         → Vng / _ -[X]
         → Ṽ   / _
```

Some examples, using /tuN-/ 'to poke', are:

```
(14)  tuN- + -boro- + -Ø- +     -na        →  tumborona
      poke-  -fully   -Ind.Tp   -Ind.S3s
      'He/she pokes all of(it)'

(15)  tuN- + -de- + -Ø- +  -na             →  tundena
      poke-  -well  -Ind.Tp-Ind.S3s
      'He/she pokes (it) well'

(16)  tuN- + -g- + -a- +   -ne-    -nga   →  tungganenga
      poke-  -TO   -Ind.Trp/Tip/Tp-Ind.S3p-SR
      'While he/she poked (it) (switch referent)... '

(17)  tuN- + -Ø- +  -na                    →  t  na
      poke-  -Ind.Tp-Ind.S3s
      'He/she pokes (it)'

(18) tuN- + -a-  +  -ne                    →  t  wane
      poke- -Ind.Trp-Ind.S3p
      'They poked (it)'
```

1.2 Morphotactics

1.2.1 Clitics

Ogea has three clitics: *-fe* 'like', *-nga* 'emphatic', and *-ha* 'pertaining to'. In native authored text, Ogeans concatenate these morphemes with the preceding word rather than writing them as separate words. The clitics *-fe, -ha,* and *-nga* appear to attach to a root of any non-verb class. *-nga* also attaches to verb roots and endocentric suffixes, as covered in section **1.2.2 Adverbial Suffix**.

1.2.1.1 -fe ‘like’

```
(19)  buuwafe
      buuwa-fe
      pig  -like
      'like a pig'
```

1.2.1.2 -nga 'just/only'

```
(20)  wahanga
      wa  -ha           -nga
      that-pertaining.to-just/only
      'only that'

(21)  nonga
      no    -nga
      he/she-just/only
      'just he/she'

(22)  yamenga
      yame-nga
      my  -just/only
      'just mine'

(23)  la   -nga
      langa
      at   -just/only
      'just there'
```

1.2.1.3 -ha ‘pertaining to’

```
(24)  waha
      wa  -ha
      that-pertaining.to
      'regarding that'

(25)  yameha
      yame-ha
      my  -pertaining.to
      'of mine'

(26)  mataneha
      matane -ha
      village-pertaining.to
      'of (the) village'

(27)  matane  anyakaroha
      matane  anyakaro-ha
      village large   -pertaining.to
      'of the large village'
```

1.2.2 Adverbial Suffix

As an adverbial suffix, *-nga* can suffix verb roots or any endocentric suffix.

```
(28)  lengawai
      le   -nga      -wa     -i
      speak-just/only-Ind.Trp-Ind.S3s
      'he/she only spoke'

(29)  letungawai
      le   -tu -nga      -wa     -i
      speak-O3s-just/only-Ind.Trp-Ind.S3s
      'he/she spoke only to him/her'
```

1.2.3 Verbs

1.2.3.1 Derived Verbs

In Ogea, it appears that any adjective may be used as a verb. Using an adjective as a verb semantically encodes a state occurring in the past or future. Present states are encoded using stative constructions which pair demonstratives, pronouns, and noun phrases with an adjective or noun phrase. (There is no copulative verb in Ogea. To say 'the man is big', Ogeans say 'the man big'). The examples, below, of adjectives used as verbs are presented with three sets for each adjective. The first part shows the adjective used in a stative construction. The second and third parts show the adjective suffixed as a verb. The third part also shows the use of the causative suffix *-mu-*.

```
(30)  a. no         tutu
         he/she/it short
         'He/she/it is short.'

      b. tutuwona
         tutu -wo     -na
         short-Ind.Tf -Ind.S3s
         'It will become short.'
```

```
        c.  tutumutuwau
            tutu -mu       -tu -wa   -u
            short-causative-O3s-Sj.S2-Sj.SS
            'Make it become short!', or 'Shorten it!'


(31)    a.  wa inyaba
            it bad
            'It is bad'

        b.  inyabawai
            inyaba -wa     -i
            bad    -Ind.Trp-Ind.S3s
            'It became bad'

        c.  inyabamutudu
            inyaba -mu       -tu -du
            bad    -causative-O3s-TS
            'Having harmed him/her…'

(32)    a.  wa        tigini
            he/she/it straight
            'He/she/it is straight'

        b.  tiginiwona
            tigini  -wo    -na
            straight-Ind.Tf-Ind.S3s
            'It will become straight'

        c.  tiginimuwai
            tigini  -mu       -wa     -i
            straight-causative-Ind.Trp-Ind.S3s
            It was made straight'
```

1.2.3.2 Reduplication

In Ogea, reduplication occurs in three forms. In one form, the first consonant and vowel (CV) of the root is reduplicated. In the second form, the entire stem is reduplicated. It is also possible that CV reduplication may be embedded in a reduplicated stem, as will be seen in the examples below. In the third form, an entire suffix is reduplicated.

1.2.3.2.1 Reduplication of the First CV of a Stem

Although Ogea has object suffixes that encode for person and number, these suffixes are not used for transitive verbs that take inanimate objects. In the case of actions involving more than one inanimate object, Ogea indicates plurality of objects by use of the suffix *-ru-*. In addition, the segments following the first CV of the root are deleted, and the first CV of the verb root is reduplicated. In this case, the reduplication possibly indicates repetitive action since more than one object is involved. Such CV reduplication is always accompanied by the plural suffix *-ru*. Note that the CV reduplication only applies to roots that begin with a consonant.

```
(33)  a. tauni
         tau  -Ø     -ni
         plant-Ind.Tp-Ind.S1/2s
         'I am planting'

      b. tataruni
         ta   -ta   -ru-Ø     -ni
         plant-plant-p2-Ind.Tp-Ind.S1/2s
                  'I am planting (many things)'

(34)  a. kouni
         kou         -Ø     -ni
         break.in.two-Ind.Tp-Ind.S1/2s
         'I am breaking into two (pieces)'

      b. kokoruni
         ko          -ko          -ru-Ø      -ni
         break.in-two-break.in.two-p2 -Ind.Tp-Ind.S1/2s
         'I am breaking (many things) into two (pieces)'

(35)  a. karuni
         karu-Ø     -ni
         cut -Ind.Tp-Ind.1/2s
         'I am cutting'
```

```
        b. kakaruni[6]
           ka -ka  -ru-Ø     -ni
           cut-cut -p2-Ind.Tp-Ind.S1/2s
           'I am cutting (many things)'
```

1.2.3.2.2 Reduplication of a Stem

In Ogea, reduplication of an entire stem indicates habitual action. As noted above, CV reduplication involving plural objects only applies to verb roots beginning with a consonant. In (39), the non-reduplicated verb begins with a vowel, and therefore a plural object is only indicated by the plural suffix *-ru*. However, the entire stem is reduplicated to indicate habitual action.

```
(36)    a. yafani
           yafa-Ø     -ni
           sit -Ind.Tp-Ind.S1/2s
           'I am sitting'

        b. yafayafani
           yafa-yafa-Ø     -ni
           sit -sit -Ind.Tp-Ind.S1/2s
           'I habitually sit'

(37)    a. karuni
           karu-Ø     -ni
           cut -Ind.Tp-Ind.S1/2s
           'I am cut'

        b. karukaruni
           karu-karu-Ø     -ni
           cut -cut -Ind.Tp-Ind.S1/2s
           'I habitually cut'

(38)    a. kakaruni
           ka -ka  -ru-Ø     -ni
           cut-cut -p2-Ind.Tp-Ind.S1/2s
           'I am cutting(many things)'

        b. kakarukakaruni
           ka -ka -ru-ka -ka -ru-Ø      -ni
           cut-cut-p2-cut-cut-p2 -Ind.Tp-Ind.S1/2s
           'I habitually cut(many things)'
```

[6] The /ru/ at the end of verb root /karu/ is not the plural marker *-ru*.

```
(39)  a. ayaruni
         aya  -ru-Ø -ni
         throw-p2 -Ind.Tp-Ind.S1/2s
         'I am throwing (many things)'

      b. ayaruayaruni
         aya  -ru-aya  -ru-Ø      -ni
         throw-p2 -throw-p -Ind.Tp-Ind.S1/2s
         'I habitually throw (many things)

(40)  a. tataruni
         ta   -ta   -ru-Ø      -ni
         plant-plant-p2 -Ind.Tp-Ind.S1/2s
         'I am planting (many things)'

      b. tatarutataruni
         ta   -ta   -ru-ta   -ta    -ru-Ø     -ni
         plant-plant-p2-plant-plant-p2-Ind.Tp-Ind.S1/2s
         'I habitually plant (many things)'
```

1.2.3.2.3 Reduplication of a Suffix

Reduplication of suffixes indicates a habitual or repeated action (the distinction depending on context)[7]. In Ogea, only endocentric suffixes are reduplicated. In the examples below, an object suffix is reduplicated (41), an manner suffix is reduplicated (42), and a benefactive suffix is reduplicated (43).

```
(41)  a. letuni
         le   -tu -Ø     -ni
         speak-O3s-Ind.Tp-Ind.S1/2s
         'I am speaking to him'

      b. letutuni
         le   -tu -tu -Ø     -ni
         speak-O3s-O3s-Ind.Tp-Ind.S1/2s
         'I spoke repeatedly to him'
```

[7] In Ogea, reduplication is but one of two mechanisms to encode habitual action. When it occurs in final verbs, the suffix *-g-* also indicates habitual action. The difference between the use of *-g-* versus reduplication for habitual action is not known, though it may be a matter of focus. For example, reduplicating an object suffix may serve to focus attention on the recipient of the action rather than the actor.

```
(42)  a. isideni
         isi -de  -Ø     -ni
         hear-well-Ind.Tp-Ind.S1/2s
         'I am hearing well'

      b. isidedeni
         isi -de  -de  -Ø     -ni
         hear-well-well-Ind.Tp-Ind.S1/2s
         'I always hear well'

(43)  a. huurufuni
         huuru-fu -Ø     -ni
         send -B3s-Ind.Tp-Ind.S1/2s
         'I send for him'

      b. huurufufuni
         huuru-fu -fu -Ø     -ni
         send -B3s-B3s-Ind.Tp-Ind.S1/2s
         'I always send for him'
```

1.2.3.3 Verb Endocenter

The structure of the Ogea endocenter is shown in Table 3. As discussed above, endocentric morphemes include the verb stem and various suffixes that may occur between the stem and the verb exocenter. Endocentric suffixes apply solely to the clause containing the verb stem being suffixed.

Table 3 Ogea Verb Endocenter

	Endocenter						
Morpheme Class	**Stem**		**Plurality**	**Object**	**Emphatic**	**Manner**	**Benefactive**
	Root	**Causative**					
Order Class	0	1	2	3	4	5	6
Required?	Yes	No	No	No	No	No	No
Values	Verb Roots Adjectives Postpositional (bagu-)	-mu-	-gana- -ru-	Table 10	-nga-	Table 11	Table 12

Examples of the causative suffix used with adjectives were given in (30) through (33), and an example of the postpositional *bagu* was given in **Error! Reference source not found.**. Examples of the causative suffix used with verb roots are given in (44) and

(45), below. Examples of the plural suffix *-ru-* were given in (33), (34), and (35). Examples of the plural suffix *-gana-* are given below in (46) and (47).

```
(44)  heigemuwai
      heige -mu       -wa     -i
      appear-causative-Ind.Trp-Ind.S3s
      'He/she/it caused (something) to appear'

(45)  usumudu
      usu   -mu       -du
      finish-causative-TS
      'He/she/it caused it to finish'

(46)  bologanawai
      bolo-gana-wa      -i
      full-p1   -Ind.Trp-Ind.S3s
      'It filled them'

(47)  dageganafuwai
      dage -gana-fu -wa      -i
      bound-p1   -B3s-Ind.Trp-Ind.S3s
      'He/she bound them for him/her'
```

The difference between *-ru-* and *-gana-* is not known at present, except that *-gana-* does not suffix stems created by reduplicating the initial CV of a root, whereas *-ru-* does. Note that *-gana-* is glossed as *p1* (plural type one) and *-ru-* as *p2* (plural type two). Examples of the emphatic suffix were given in (28) and (29). In the case of suffix values that have a table number instead of an actual value, examples of the suffixes follow the specified table.

1.2.3.4 Verb Exocenters

As stated in the introduction to this appendix, Ogea verb exocenters include suffixes that encode inter-clausal temporality, tense, mood, subject, and switch reference. They are termed exocentric because they may contain two suffixes that relate to the clause that follows (i.e., inter-clausal temporality and switch reference relate the current clause to the one that follows it). Ogea verb exocenters fall into the following categories: indicative, contrafactual, subjunctive, and infinitive. Each is discussed in turn, below.

1.2.3.4.1 Indicative Exocenters

The indicative exocenters are broken into two major categories: sentence *medial* and sentence *final* forms. The medial exocenters are further categorized into *same* and *different* referents.

The medial exocenters include temporality suffixes. Temporality suffixes are medial verb suffixes that indicate a temporal relationship between the current clause and the one following it. Actions can occur at the same time (temporal overlap) or in sequence (temporal succession). When comparing the tables, below, for the indicative endocenters, note the following. Temporal succession is *marked* for same referent verbs and *unmarked* for different referent verbs. Temporal overlap is *marked* for different referent verbs and *unmarked* for same referent verbs. Also note that the different referent temporal overlap suffix -*g*- can appear as a suffix on final verbs, in which case it indicates habitual action rather than temporal overlap.

1.2.3.4.1.1 Medial, Same Referent, Indicative Exocenter

The medial, same referent, indicative exocenter has but one suffix. The presence of the succession temporality suffix -*du* indicates that the action or state of the verb suffixed by -*du* is completed before the action or state of the subsequent verb. The absence of the succession temporality suffix -*du*, i.e. the occurrence of the zero morpheme -Ø, indicates that the action of the verb overlaps in time with the action of the subsequent verb.

Table 4 Ogea Medial, Same Referent, Indicative Verb Exocenter

	Exocenter	
Suffix Class	**Temporality**	
Order Class	7	
Required?	Yes	
Values	Succession	-du
	Overlap	-Ø

```
(48)  no        matane  awadu    fagawai
      no        matane  awa  -du faga    -wa     -i
      he/she/it village leave-TS run.away-Ind.Trp-Ind.S3s
      'Having left the village, he/she/it ran away.'

(49)  no        nere beele beneu     leniridu
      no        nere beele be  -neu  le   -niri-du
      he/she/it them words this-like speak-O3d -TS

      bona           huurunirai
      bo   -na       huuru-nir-a      -i
      vAuxR-vAux.S3s send -O3d-Ind.Trp-Ind.S3s

      'Having said this to them, he/she/it sent them off.'

(50)  no        jaidu     bona
      no        jai   -du bo   -na
      he/she/it get.up-TS vAuxR-vAux.S3s

      otodu    golowai
      oto  -du golo-wa     -i
      stand-TS walk-Ind.Trp-Ind.S3s

      'Having stood up, he/she/it walked.'

(51)  no        yafa   bona           lewai
      no        yafa-Ø bo   -na       le   -wa     -i
      he/she/it sit-TO vAuxR-vAux.S3s speak-Ind.Trp-Ind.S3s
      'As he/she/it sat, he/she/it spoke.'

(52)  no        fele     bona           lewai
      no        fele-Ø   bo   -na       le   -wa     -i
      he/she/it watch-TO vAuxR-vAux.S3s speak-Ind.Trp-Ind.S3s
      'As he/she/it watched, he/she/it spoke.'
```

Note that medial, same referent verbs are often followed by the verbal auxiliary *bo-*, discussed in a separate section, below.

1.2.3.4.1.2 Medial, Different Referent, Indicative Exocenter

Whereas the medial, same referent, indicative exocenter is compact, the medial, different referent, indicative exocenter is the opposite. Also, the medial, different referent, indicative exocenter contains suffixes that overlap with those of the final, indicative exocenter. However, the two exocenters differ in two ways. A different meaning is encoded by the initial suffix *-g-*, and the final suffixes differ. The initial suffix *-g-* encodes temporality in the case of the medial, indicative exocenter, but encodes habituality in the case of the final, indicative exocenter. The suffix *-nga* is the final suffix of the medial, same referent, indicative exocenter, and the suffix *-ha* is the final suffix of the final, indicative exocenter. Therefore, except for the final suffix, both the medial, different referent, indicative exocenter, and the final, indicative exocenter have the same form.

The presence of the succession temporality suffix zero morpheme indicates that the action or state of the verb is completed before the action or state of the subsequent verb. See (55) for an example. The succession temporality suffix *-g-* indicates that the action or state of the verb overlaps in time with the action or state of the subsequent verb. An example of overlap in state rather than action is seen in (51).

Note that for the medial, same referent exocenter, the switch referent suffix is mandatory.

Table 5 Ogea Medial, Different Referent, Indicative Exocenter

	Exocenter				
Suffix Class	**Temporality**		**Tense**	**Subject**	**Switch Referent**
Order Class	7		8	9	11
Required?	Yes		Yes	Yes	Yes
Values	Succession	-Ø-	Table 16	Table 17	-nga
	Overlap	-g-			

```
(53)  no onou lenigigainga
      no onou le   -nigi-g -a             -i      -nga
      he thus speak-O3p -TO-Ind.Trp.Tip/Tp-Ind.S3s-SR

      fai baingaro medu    bode
      fai baingaro me  -du bo   -de
      man many     come-TS vAuxR-vAux.S3p

      fai waha bagu otowane
      fai waha bagu oto  -wa     -ne
      man that with stand-Ind.Trp-Ind.S3p

      'While he spoke thus to them, many men came and stood with
       that man.'

(54)  ere nere awanigigeyenga
      ere nere awa  -nigi-g -e      -ye     -nga
      we  they leave-O3p -TO-Ind.Trp-Ind.S1p-SR

      nere ewe  onou      inyiinyide
      nere ewe  onou      inyi-inyi-ø            -de
      they idle like.that be  -be  -Ind.Tp-Ind.S3p

      'They remain idly abandoned by us.'

(55)  nere no        huurutuwanenga
      nere no        huuru-tu -ø   -wa            -ne     -nga
      they he/she/it send -O3s-TS-Ind.Trp/Tip/Tp-Ind.S3p-SR

      matane  langa  minai
      matane  la-nga min    -a      -i
      village at-int go.down-Ind.Trp-Ind.S3s

      'They sent him/her/it, and he/she/it went down to the
       village.'
```

In (53), the verb for 'speak' (in the first clause of the sentence) is marked for switch reference, since the referent of the first clause (an individual man) is different than the referents of the subsequent clauses (a group of men).

1.2.3.4.1.3 Final, Indicative Exocenter

The final, indicative exocenter is used with a verb occurring in the final clause of a sentence. However, the final form verb may occur in a non-sentence final position if it is part of a sentence embedded as a relative clause[8]. It is none-the-less the final verb of the embedded sentence. An example of this is seen in (58).

Table 6 Ogea Final, Indicative Exocenter

	Exocenter			
Suffix Class	**Habituality**	**Tense**	**Subject**	**Perfective**
Order Class	7	8	9	10
Required?	No	Yes	Yes	No
Values	-g-	Table 16	Table 17	-ha

An example of the habitual suffix is (56) and of the perfective suffix is (57).

```
(56)  ou    hogo fere      fere      gaigai
      ou    hogo fere      fere      gaigai
      heavy face different different always

      ere bagu heigegai
      ere bagu heige -g  -a      -i
      we  with appear-Hab-Ind.Trp-Ind.S3s

      'Various kinds of problems repeatedly happened to us.'

(57)  abitimutuwaneha
      abiti -mu       -tu -wa         -ne     -ha
      show  -causative-O3s-Ind.Trp/Tip-Ind.S3s-Perf
      'They have already caused (it to be) shown to him/her'
```

[8] Structurally, Ogea relative clauses are formed by embedding a sentence into the head noun slot of a definite noun phrase. See Colburn [2].

```
(58)  fai dada  nere nage       owonege  la
      fai dada  nere nage       owo-nege la
      man thing they themselves hand-P3p with

      tafade              waha
      tafa-Ø     -de      wa  -ha
      put -Ind.Tp-Ind.S3p that-perf

      ada mogetude
      ada moge        -tu -Ø     -de
      not turn.back.on-O3s-Ind.Tp-Ind.S3p

      'Men do not turn their backs on that which they themselves
      have made.'
```

1.2.3.4.2 Contrafactual Exocenter

The use of contrafactual suffixes indicates the speaker's belief that an event did not actually occur.

Table 7 Ogea Contrafactual Verb Exocenter

	Exocenter				
Suffix Class	**Temporality**		**Subject**	**Contrafactual**	**Switch Reference**
Order Class	7		8	10	11
Required?	Yes		Yes	Yes	No
Values	Succession	-Ø-	Table 19	-ge-	-nga
	Overlap	-g-			

Following are examples of Ogea contrafactual clauses:

```
(59)  aririmuhagage
      ariri  -mu       -hag-a              -ge
      deflect-causative-B1p-Contra.2/3S/d/p-Contra
      'if they had caused it to be deflected for us…'

(60)  ji yafegenga
      ji yaf-ø  -e         -ge    -nga
      I  sit-TS-Contra.S1s-Contra-SR
      'if I had sat…'
```

1.2.3.4.3 Subjunctive Exocenter

The term 'subjunctive' is used loosely here. Traditional usage of the term subjunctive contrasts it with the imperative and other moods. In Ogea, the subjunctive and imperative are not differentiated. The suffixes of the Ogea subjunctive exocenter (see Table 8) encode the deontic mood (obligation and permission[9]) as well as the predictive. As such, the subjunctive suffixes are translated by a variety of English modal auxiliaries (e.g., must, might, will). In the absence of the predictive and intensifier suffixes, the subjunctive person and number suffixes encode the deontic mood (i.e., *I must, you must, he/she must*, etc.). When used with second person subject suffixes, the deontic mood encodes what would in English be called the imperative mood.

Table 8 Ogea Subjunctive Exocenter

	Exocenter			
Morpheme Class	**Subject Person**	**Subject Number**	**Predictive**	**Intensifier**
Order Class	8	9	10	11
Required?	Yes	Yes	No	No
Value	Table 20	Table 21	-ga-	-nga

By adding the predictive suffix, the speaker is stating a belief that an event is likely to happen. This assertion may be further strengthened by the addition of the intensifier suffix. (In this author's original analysis, *-ga-* was analyzed as encoding apprehension. Apprehension is not contradictory to prediction, but does not capture the full usage of *-ga-*. Also, when clearly indicating apprehension, the word *gamaji* 'lest' will occur in the clause.) Below are examples of Ogea subjunctive clauses. More examples are provided with Table 20 and Table 21.

```
(61)  yafadewaguganga
      yafa-de  -wa   -gu   -ga  -nga
      sit -well-Sj.S2-Sj.Sp-pred-int
```

[9] In and of themselves, the Ogea subjunctive suffixes do not encode permission. A clause encoding permission must have both a verb with subjunctive suffixes and the word *dabu* 'may'.

```
      boni           leni
      bo    -ni      le   -Ø          -ni
      vAuxR-vAux.S1s speak-Ind.Tp-Ind.S1s

      'I say this that you might have a good life'

(62)  tere       huwanyatege  ngalengawaguga
      tere       huwanya-tege ngalenga-wa   -gu   -ga
      you.plural inside -P3p  true    -Sj.S2-Sj.Sp-pred

      'that you might believe…'
```

1.2.3.4.4 Infinitive Exocenter

The term 'infinitive' is used in this author's analysis to refer to verbs whose exocenter only encodes subject number (singular, dual, and plural). Such verbs lack suffixes for temporal succession, tense, switch reference, or subject person. Examples are given following Table 9 and Table 22.

Table 9 Ogea Infinitive Exocenter

	Exocenter	
Morpheme Class	**Infinitive**	**Subject**
Order Class	8	9
Required?	Yes	Yes
Value	-e-	Table 22

```
(63)  umeruwegu         uwau
      ume-ru-we -gu     uwau
      die-p2-Inf-Inf.Sp without
      'without dying'

(64)  tere       ji awahiyegu           uuwa
      tere       ji awa  -hiy-e  -gu     uuwa
      you.plural I  leave-O1s-Inf-Inf.Sp not
      'you did not leave me'
```

```
(65)  haifuwegu           nomo bona
      hai -fu -we -gu     nomo bo   -na
      tell-B3s-Inf-Inf.Sp its  vAuxR-vAux.S3s
      'in order to tell on his behalf'

(66)  goloweru        nomo edo  uuwa
      golo-we -ru     nomo edo  uuwa
      walk-Inf-Inf.Sd its  able not
      'they (dual) were not able to walk'

(67)  yeregei          nomo bode
      yereg-e  -i      nomo bo   -de
      write-Inf-Inf.Ss its  vAuxR-vAux.S3p
      'in order that they could write'

(68)  lalanigei            nomo bona
      lala -nig-e  -i      nomo bo   -na
      light-O3p-Inf-Inf.Ss its  vAuxR-vAux.S3s
      'in order to give light to them'
```

Note that each of the above examples with *bo-* can be transformed to encode an inclusive plural subject. This is done by 1) use of the first person plural subject pronoun, 2) omission of *bo-*, and 3) ending the sentence with an infinitive verb plus *nomo*. For example, (69), a modification of (68), encodes an inclusive plural, whereas (70) encodes an exclusive plural.

```
(69)  ere lalanigei           nomo
      ere lala -nig-e  -i     nomo
      we light-O3p-Inf-Inf.Ss its
      'Let us (inclusive) give light to them'

(70)  ere lalanigoya
      ere lala -nig-o     -ya
      we light -O3p-Ind.Tf-Ind.S1p
      'We(exclusive) will give light to them'
```

1.2.3.5 Verbal Suffixes

1.2.3.5.1 Object Suffixes

Table 10 Ogea Object Suffixes

		Number		
		Singular	**Dual**	**Plural**
Person	**1st**	-hi-	-hiri-	-higi-
	2nd	-hini-	-tiri-	-tigi-
	3rd	-tu-	-niri-	-nigi-

There is some evidence that the underlying form of the first person singular object suffix *-hi-* is actually *-hii-*. This underlying form is hypothesized based on morphophonemic Rule 2 (discussed above) and as shown in example (71), where /-hii-/ becomes /-hiy-/. Also, with the exception of third person singular (which often has irregular forms in Ogea), if the underlying form of *-hi-* is actually *-hii-*, then there is a rule to explain the relationship between the object suffixes and the benefactive suffixes. Namely, /Ci/ → /Ca/. For these reasons, the underlying form of the first person singular object suffix *-hi-* is hypothesized to be *-hii-*. Examples (71) through (80) are paradigmatic in that they all use the same root *le-*, and show the object suffix encoding indirect objects. Examples (81) and (82) show the object suffix encoding direct objects.

```
(71)  lehiyai
      le- +  -hii- + -a- +     -i         → lehiyai
      speak- -O1s-   -Ind.Trp- -Ind.S3s
      'He/she spoke to me'

(72)  lehide
      le   -hii-Ø     -de
      speak-O1s-Ind.Tp-Ind.S3p
      'They are speaking to me'
```

```
(73)  lehinina
      le   -hini-Ø     -na
      speak-O2s -Ind.Tp-Ind.S3s
      'he/she is speaking to you (sing)'

(74)  letugai
      le   -tu -g  -a      -i
      speak-O2s-Hab-Ind.Trp-Ind.S3s
      'he/she habitually speaks to him/her'

(75)  lehiri
      le   -hiri-Ø
      speak-O1d -TO
      'while speaking to us (dual)…'

(76)  letirinaha
      le   -tiri-Ø     -na     -ha
      speak-O2d -Ind.Tp-Ind.S3s-perf
      'he/she has spoken to you (dual)'

(77)  lenirona
      le   -nir-o     -na
      speak-O3d-Ind.Tf-Ind.S3s
      'he/she will speak to them (dual)'

(78)  lehigona
      le   -hig-o     -na
      speak-O1p-Ind.Tf-Ind.S3s
      'he/she will speak to us (plural)'

(79)  letigini
      le   -tigi-Ø     -ni
      speak-O2s -Ind.Tp-Ind.S1/2s
      'I am speaking to you (plural)'

(80)  lenigini
      le   -nigi-Ø     -ni
      speak-O2s -Ind.Tp-Ind.S1/2s
      'I am speaking to them (plural)'

(81)  nuutuwona
      nuu  -tu -wo    -na
      shove-O3s-Ind.Tf-Ind.S3s
      'he/she will shove it'

(82)  hoyotudu
      hoyo-tu -du
      help-O3s-TS
      'having helped him/her…'
```

1.2.3.5.2 Manner Suffixes

Ogea manner suffixes modify verbs by stating the scope of an action—whether the action was complete in its effect, was done well, or done quickly.

Table 11 Ogea Manner Suffixes

Meaning	**Suffix**
completely	-boro-
well	-de-
quickly	-ya-

```
(83)  ayaborowai
      aya  -boro      -wa     -i
      throw-completely-Ind.Trp-Ind.S3s
      'he/she threw (all of it) away'

(84)  fiideganenga
      fii   -de  -g -a      -ne     -nga
      smooth-well-TO-Ind.Trp-Ind.S3p-SR
      'while they smoothed (it out) well…'

(85)  owonege   ayayadu
      owo -nege aya  -ya     -du
      hand-P3p  throw-quickly-TS
      'having quickly moved their hands…'
```

1.2.3.5.3 Benefactive Suffixes

Table 12 Ogea Benefactive Suffixes

		Number		
		Singular	**Dual**	**Plural**
Person	**1st**	-hai-	-hari-	-hagi-
	2nd	-hani-	-tari-	-tagi-
	3rd	-fu-	-nari-	-nagi-

As noted above in the discussion of Ogea object suffixes, there appears to be a systematic relationship between the object suffixes and the benefactive suffixes. Assuming that the first person singular object suffix *-hi-* has an underlying form of /-hii-/, then for all persons except third person singular, the initial /Ci/ becomes /Ca/ to encode benefaction. For third person singular, the initial /t/ becomes /f/.

Below are examples of Ogea benefactive suffixes. In the corpora available to this author, there were no examples of first or second person dual benefactive suffixes.

```
(86)  haiyahaidu
      haiya  -hai-du
      prepare-B1s-TS
      'having prepared it for me…'

(87)  uuwamuhaninaha
      uuwa-mu       -hani-Ø     -na     -ha
      not -causative-B2s -Ind.Tp-Ind.S3s-perf
      'he/she has caused it to cease on your behalf'

(88)  uuwamufuwai
      uuwa-mu       -fu -wa     -i
      not -causative-B3s-Ind.Trp-Ind.S3s
      'he/she caused (it) to cease on his/her behalf'

(89)  fuutunaridu
      fuutu-nari-du
      untie-B3d -TS
      'having untied (it) for them (dual)…'

(90)  jaimuhagihagina
      jai   -mu       -hagi-hagi-Ø     -na
      get.up-causative-B1p -B1p -Ind.Tp-Ind.S3s
      'he/she habitually raises (it up) for us (plural)'

(91)  feutagidu
      feu     -tagi-du
      pour.out-B2p -TS
      'having poured (it) out for you (plural)…'

(92)  aawanagidu
      aawa -nagi-du
      leave-B3p -TS
      'having left (it) for them (plural)…'
```

In Ogea, the benefactive suffixes are used both to encode benefaction and malefaction. The action performed may be for the good or the harm of an individual. An example of a malefactive usage is (93) (from a text in the Erima dialect of Ogea).

```
(93)  gaigu tehaita
      gaigu te -hai-Ø      -ta
      thigh get-B1s-Ind.Tip-Ind.S3s
      'it has gotten (my) thigh for me'
```

The context of (93) is as follows. A man is trapped in a cave filling with water. He is in immediate danger of drowning. He yells to someone outside that the water has gotten his thigh, meaning that the water has come up to his thigh. In this context, the suffix -*hai*- clearly has a malefactive meaning.

1.2.3.5.4 Indicative Verb Remote Past Tense Suffixes

The remote past refers to actions that occurred before today, and the immediate past tense refers to actions that occurred today.

Table 13 Ogea Remote Past Tense Suffixes

Person	**Suffix**
1st and 2nd	-e-
3rd	-a-

```
(94)  ne       ji huuruhiyenenga
      ne       ji huuru-hii-e             -ne       -nga
      you.sing I  sent -O1s-Ind.Trp/Tip/Tp-Ind.S1/2s-SR

      meene
      me       -e      -ne
      come.down-Ind.Trp-Ind.1/2s

      'You sent me and I came down.'

(95)  ere nere awanigigeyenga
      ere nere awa  -nigi-g -e      -ye       -nga
      we  they leave-O3p -TO-Ind.Trp-Ind.S1/2p-SR

      nere ewe  onou inyiinyide
      nere ewe  onou inyi-inyi-Ø     -de
      they idle thus be  -be  -Ind.Tp-Ind.S3p

      'We left them, and (now) they are always idle.'
```

```
(96)  onowainga                      nere
      ono    -Ø -wa     -i      -nga nere
      Do.thus-TS-Ind.Trp-Ind.S3s-SR  they

      no        letuwane
      no        le   -tu -wa         -ne
      he/she/it speak-O3s-Ind.Trp/Tip-Ind.S3p

      'Having done thus, they spoke to him/her/it.'

(97)  nere we         nere huuruniganenga
      nere we         nere huuru-nig-a             -ne     -nga
      they themselves they send -O3p-Ind.Trp/Tip/Tp-Ind.S3p-SR

      mayane
      mai     -a          -ne
      come.up-Ind.Trp/Tip-Ind.S3p

      'They themselves sent them and they came up.'

(98)  nere fai oojotugarenga
      nere fai oojo  -tu -g -a      -re     -nga
      they man follow-O3s-TO-Ind.Trp-Ind.S3d-SR

      nerirai
      nerir        -a      -i
      see.O3d.dual-Ind.Trp-Ind.S3s

      'While they were following the man, he saw them.'

(99)  isene
      is  -e      -ne
      hear-Ind.Trp-Ind.S1/2s
      'I/you (singular) heard'

(100) onowere
      ono     -we       -re
      do.thus-Ind.Trp-Ind.S1d
      'we/you (plural) did thus'

(101) goyeye
      goy-e      -ye
      go -Ind.Trp-Ind.S1/2p
      'we/you (plural) went'

(102) goyai
      goy-a      -i
      go -Ind.Trp-Ind.S3s
      'he/she went'

(103) elegare
      eleg  -a      -re
      gather-Ind.Trp-Ind.S3d
      'they (dual) gathered'
```

```
(104) goyane
      goy-a          -ne
      goy-Ind.Trp/Tip-Ind.S3p
      'they (plural) went'
```

1.2.3.5.5 Indicative Verb Immediate Past and Present Tense Suffixes

In Ogea, the immediate past tense refers to actions that occurred today. However, there are times in Ogea discourse where the immediate past tense is used in reference to events that occurred in the distant past. This appears to be a stylistic usage to more fully engage the listener in the narrative.

The present tense not only encodes an action occurring at the present point of time, but is also used when encoding a habitual action that occurred not only in the past, but continues to occur in the present, as shown in (111).

Table 14 Ogea Immediate Past and Present Tense Suffixes

Person	**Number**	**Sentence Medial**		**Sentence Final**	
		Immediate Past and Present		**Immediate Past**	**Present**
		Temporal Overlap	**Temporal Succession**		
1^{st} & 2^{nd}	**All**	-e-	-Ø-		
3^{rd}	**Singular**	-a-			
	Dual				
	Plural				

```
(105) nere soonigiganga
      nere soo  -nigi-g -a      -Ø      -nga
      they chase-O3p -TO-Ind.Tip-Ind.S3s-SR

      ureya
      ure    -Ø     -ya
      see.O3s-Ind.Tp-Ind.S1/2p

      'We see them chase them (away).'

(106) ne       aita  waha
      ne       aita  wa  -ha
      you.sing woman that-pertaining.to
      awatuninga
      awa  -tu -Ø         -ni       -nga
      allow-O3s-Ind.Tip/Tp-Ind.S1/2s-SR

      ne       ngare     yafana
      ne       ngare     yafa-Ø     -na
      you.sing with.dual sit -Ind.Tp-Ind.S3s

      'You are allowing that woman to be with you.'

(107) ere fai beha
      ere fai be-ha
      we  man this-pertaining.to

      onotuyarenga                         golona
      ono    -tu -Ø         -yare     -nga golo-Ø     -na
      do.thus-O3s-Ind.Tip/Tp-Ind.S1/2d-SR  go  -Ind.Tp-Ind.S3s

      'We did thus to this man and he is walking.'

(108) no     ji haruwe waha
      no     ji haruwe wa  -ha
      he/she I  work   that-pertaining.to

      hiyanga
      hii     -a         -Ø      -nga
      give.O1s-Ind.Tip/Tp-Ind.S3s-SR

      ji haruwe waha               teni
      ji haruwe wa-ha              te  -Ø         -ni
      I  work   that.pertaining.to take-Ind.Tip/pT-Ind.S1/2s

      'I do the work he/she gives me.'


(109) tere        taate boya             nere
      tere        taate bo   -ya         nere
      you.plural  what  vAuxR-vAux.S1/2p they

      awanigiyanga                        nere
      awa  -nigi-Ø         -ya       -nga nere
      allow-O3p -Ind.Tip/Tp-Ind.S1/2p-SR  they

      inyabamutigegu                  uwa
      inyaba-mu       -tig-e  -gu     uwa
      bad   -causative-O2p-Inf-Inf.Sp not

      'Why did you not allow them to harm you?' or
      'Why do you not allow them to harm you?'

(110) ji uulufuni
      ji uulu    -fu -Ø     -ni
      I  peel.off-B3s-Ind.Tp-Ind.S1/2s
      'I peel (it) off for he/she/it'
```

```
(111) ne       ada tafaniginigini
      ne       ada tafa-nigi-nigi-Ø     -ni
      you.sing not put -O3p -O3p -Ind.Tp-Ind.S1/2s
      'You do not habitually put them'
```

1.2.3.5.6 Future Tense Suffixes

Ogea future tense suffixes are shown in Table 15 and examples are provided below. Two things should be noted about Ogea final form future tense verbs. First, for the future tense, second person subjects are encoded using the subjunctive forms, so final form future tense suffixes are not applicable to the indicative future tense paradigm. However, they are shown anyway to capture the insight that 2nd person medial and final tenses are marked identically. Second, the 1st and 3rd person subject suffixes have allomorphic variants that optionally occur when following a future tense final form suffix, and when preceding the verbal auxiliary *bo-* 'intent'. The variants are optional in that there is variation both within a single speaker's speech and across speakers. The variation may be stylistic. The variants are as follows:

```
(112) lewoni + boni     → lewon boni    (loss of final vowel)
      le   -o     -ni      bo   -ni
      speak-Ind.Tf-Ind.S1s vAuxR-vAux.S1s
      'I will speak'

(113) lewona  + bona    → lewon bona    (loss of final vowel)
      le   -o     -na      bo   -na
      speak-Ind.Tf-Ind.S3s vAuxR-vAux.S3s
      'He/she will speak'

(114) lewoyare + boyare → lewo  boyare (loss of suffix)
      le   -o     -yare     bo   -yare
      speak-Ind.Tf-Ind.S1d  vAuxR-vAux.S1d
      'We.dual will speak'

(115) lewodere + bodere → lewod bodere (loss of final VCV)
      le   -o     -dere     bo   -dere
      speak-Ind.Tf-Ind.S3d  vAuxR-vAux.S3d
      'They.dual will come'
```

```
(116) lewoya   + boya   → lewo boya   (loss of suffix)
      le   -o     -ya      bo   -ya
      speak-Ind.Tf-Ind.S1p vAuxR-vAux.S1p
      'We.plural will speak'

(117) lewode   + bode   → lewod bode    (loss of final vowel)
      le   -o     -de   +   bo   -de
      speak-Ind.Tf-Ind.S3p  vAuxR-vAux.S3p
      'They.plural will speak'
```

Note that in some cases the final V or the final VCV is lost, and in other cases the entire subject suffix is lost. Also note that the loss of either the entire subject suffix or the final V or final VCV results in ambiguity for a morphological parser that cannot look ahead to the next word to disambiguate the subject suffix. For example, in (126), below, the subject for the verb 'give' is ambiguous between 1st and 3rd person unless information can be obtained by looking ahead to the next word, or by obtaining information from a syntactic parser that identifies the 1st person pronoun *ji* as being the grammatical subject. Note that these variants provide the second exception to the general rule in Ogea that syllables are open rather than closed. In the examples below, future tense verbs are shown with both the normal subject suffix, and the forms that optionally occur when preceding the verbal auxiliary *bo-*.

Table 15 Ogea Future Tense Suffixes

Person	**Sentence Position**	
	Medial	**Final**
1st	-e-	-o-
2nd	-a-[10]	
3rd	-o-	

[10] Technically the 2nd person future sentence final tense does not belong in a table for the indicative future tenses. However, it is shown here to capture the insight that both the indicative 2nd person medial future tense and the 2nd person subjunctive tense are marked identically.

```
(118) ji no        huurutuwehenga
      ji no        huuru-tu -e     -he     -nga
      I  he/she/it send -O3s-Ind.Tf-Ind.S1s-SR

      tere bagu mona                     bona
      tere bagu m        -o     -na      bo   -na
      you  with come.down-Ind.Tf-Ind.S3s vAuxR-vAux.S3s
      'I will send him/her/it to you.plural and
      he/she/it will come'

(119) ne fasadu   mayahenga
      ne fasadu   mai    -a     -he     -nga
      you quickly come.up-Ind.Tf-Ind.S2s-SR

      minoyare
      min    -o     -yare
      go.down-Ind.Tf-Ind.S1/2d

      'You come up quickly and we will go down.'

(120) ere megebu ayarudu     waredenga
      ere megebu aya  -ru-du war    -e     -de     -nga
      we  stone  throw-p2-TS hit.O3s-Ind.Tf-Ind.S1p-SR

      umona              bona
      um -o     -na      bo   -na
      die-Ind.Tf-Ind.S3s vAuxR-vAux.S3s

      'We will throw stones and hit him/her and he/she will die.'

(121) tere       nere fai ila  adai   awanigadenga
      tere       nere fai ila  adai   awa  -nig-a     -de    -nga
      you.plural they man some do.not allow-O3p-Ind.Tf-Ind.S2p-SR

      nere yaawatigode
      nere yaawa-tig-o     -de
      they lie  -O2p-Ind.Tf-Ind.S3p

      'Do not allow any men to deceive to you.'

(122) ne         taate  dada  bonahe                 gau
      ne         taate  dada  bo   -nahe             gau
      you.plural what   thing vAuxR-vAux.vAux.Tf.S2s desire

      hinonga                    ji
      hin     -o     -Ø      -nga ji
      give.O2s-Ind.Tf-Ind.S3s-SR  I

      hinon                      boni
      hin     -o     -n          bo   -ni
      give.O2s-Ind.Tf-Ind.S1/3s vAuxR-vAux.S1s

      'Whatever you desire, I will give you.'
```

```
(123) nere haruwe usumuworonga
      nere haruwe usu   -mu       -wo    -ro     -nga
      they work   finish-causative-Ind.Tf-Ind.S3d-SR

      kaafa kui   ete no        mai     heigedu   bona
      kaafa kui   ete no        mai     heige -du bo   -na
      animal wild one he/she/it come.up arrive-TS vAuxR-vAux.S3s

      nere fai waha               bagu
      nere fai wa   -ha           bagu
      they man that-pertaining.to with.plural

      warawona             bona
      wara -wo    -na      bo   -na
      fight-Ind.Tf-Ind.S3s vAuxR-vAux.S3s

      'After they.dual have finished working, a wild animal will
      come up and fight with them.dual.'

(124) nere ji yarogonga
      nere ji yar    -o     -go     -nga
      they I  hit.O1s-Ind.Tf-Ind.S3p-SR

      umon                 boni
      um -o     -n         bo   -ni
      die-Ind.Tf-Ind.S1/3s vAuxR-vAux.S1s

      'They will strike me dead.'

(125) awatuwoni
      awa  -tu -wo    -ni
      leave-O3s-Ind.Tf-Ind.S1s
      'I will leave him/her/it'

(126) ji ne       hinon                     boni
      ji ne       hin     -o     -n         bo   -ni
      I  you.sing give.O2s-Ind.Tf-Ind.S1/3s vAuxR-vAux.S1s
      'I will give (it) to you'

(127) no yafona
      no yaf-o     -na
      he sit-Ind.Tf-Ind.S3s
      'He will sit'

(128) lenigon                     bona
      le   -nig-o      -n         bo   -na
      speak-O3p-Ind.Tf-Ind.S1/3s vAuxR-vAux.S3s
      'he/she/it will speak to them'

(129) minoyare
      min    -o     -yare
      go.down-Ind.Tf-Ind.S1d
      'We.dual will go down'
```

```
(130) letigo                     boyare
      le-tig   -o     -Ø         bo   -yare
      speak-O2p-Ind.Tf-Ind.S1d/p vAuxR-vAux.S1d
      'We.dual will speak to you.plural'

(131) umeruwodere
      ume-ru-wo     -dere
      die-p2 -Ind.Tf-Ind.S3d
      'They.dual will die'

(132) heigod                  bodere
      heig  -o     -d         bo   -dere
      appear-Ind.Tf-Ind.S3d/p vAuxR-vAux.S3d
      'They.dual will appear'

(133) nyoya
      ny -o     -ya
      eat-Ind.Tf-Ind.S1p
      'We will eat'

(134) nyo                  boya
      ny -o     -Ø         bo   -ya
      eat-Ind.Tf-Ind.S1d/p vAuxR-vAux.S1p
      'We will eat'

(135) tafatuwode
      tafa-tu -wo     -de
      put -O3s-Ind.Tf-Ind.S3s
      'They will put him/her/it'

(136) tod                  bode
      t  -o     -d         bo   -de
      get-Ind.Tf-Ind.S3d/p vAuxR-vAux.S3p
      'They will get (it)'
```

1.2.3.5.7 Summary of Indicative Verb Tense Suffixes

This section provides a summary table for the Ogea indicative tense suffixes. The abbreviations follow the standard ones used throughout this appendix: Remote Past Tense (RPT), Immediate Past Tense (IPT), Present Tense (PT), Future Tense (FT), Temporal Overlap (TO), and Temporal Succession (TS). Sentence Medial forms are for those involving switch reference. Medial forms for the same referent were given above in section *1.2.3.4.1.1 Medial, Same Referent, Indicative Exocenter.*

Table 16 Ogea Indicative Tense Suffixes - Summary

Person	Number	Sentence Medial						Sentence Final			
		RPT		IPT & PT		FT		RPT	IPT	PT	FT
		TO	TS	TO	TS	TO	TS				
1st	All	-e-			-Ø-	-e-			-Ø-		-o-
2nd						-a-					-a-[11]
3rd	Singular	-a-				-o-		-a-			-o-
	Dual										
	Plural										

1.2.3.5.8 Indicative Verb Subject Suffixes

Below is a table for the Ogea indicative verb subject suffixes. The abbreviations used in the table are the same ones used throughout the appendix and described in the previous section. The sentence final subject suffixes shown for the future tense do not apply to the 2nd person because 2nd person future suffixes are encoded by subjunctive forms (see Table 8). Note that in both sentence medial and final positions the remote past tense subject suffixes are identical.

Table 17 Ogea Indicative Verb Subject Suffixes

Person	Number	Sentence Medial						Sentence Final			
		RPT		IPT & PT		FT		RPT	IPT	PT	FT[12]
		TO	TS	TO	TS	TO	TS				
1st & 2nd	Singular	-ne-[13]				-he-		-ne-	-na-	-ni-	
	Dual	-re-			-yare-	-dere-		-re-	-yare-		
	Plural	-ye-			-ya-	-de-		-ye-	-ya-		
3rd	Singular	-i-	-Ø-					-i-	-Ø-	-na-	
	Dual	-re-				-ro-		-re-		-dere-	
	Plural	-ne-				-go-		-ne-		-de-	

[11] Technically the 2nd person future sentence final tense does not belong in a table for the indicative tenses. However, it is shown here to capture the insight that both the indicative 2nd person medial future tense and the 2nd person subjunctive tense are marked identically.

[12] 2nd person does not apply to sentence final indicative future tense. The subjunctive forms are used instead: -a-, -ru-, -gu-.

Examples of indicative verb subject suffixes occur throughout the appendix. The examples given with the various indicative tenses above also provide examples of each subject suffix in combination with each tense suffix.

1.2.3.5.9 Summary of Indicative Verb Tense and Subject Suffixes

By now the reader may have noticed that in isolation, there often is ambiguity in the Ogea tense suffixes. For example, the tense suffix *-e-* is used for both the remote past (in the case of 1st and 2nd person), and in the future (in the case of 1st person medial verbs). In isolation, the indicative verb subject suffixes also have some ambiguity. When the tense and subject suffixes are combined, some, but not all, ambiguity is resolved. For the reader's convenience, a summary table is presented below.

Table 18 Ogea Indicative Verb Tense and Subject Suffixes Combined

Prsn	Nmbr	Sentence Medial						Sentence Final			
		RPT		IPT & PT		FT[14]		RPT	IPT	PT	FT[15]
		TO	TS	TO	TS	TO	TS				
1st & 2nd	Sing	-e-ne-			-Ø-ne-	-e-he- -a-he-		-e-ne-	-Ø-na-	-Ø-ni-	-o-ni-
	Dual	-e-re-			-Ø-yare-	-e-dere- -a-dere-		-e-re-	-Ø-yare-		-o-yare-
	Plural	-e-ye-			-Ø-ya-	-e-de- -a-de-		-e-ye-	-Ø-ya-		-o-ya-
3rd	Sing	-a-i-		-a-Ø-	-Ø-Ø-	-o-Ø-		-a-i-	-Ø-Ø-	-Ø-na-	-o-na-
	Dual	-a-re-				-o-ro-		-a-re-		-Ø-dere-	-o-dere-
	Plural	-a-ne-				-o-go-		-a-ne-		-Ø-de-	-o-de-

Note that the sentence medial subject affixes are always followed by *-nga*, the switch referent suffix. This resolves the ambiguity between the two occurrences of *-Ø-*

[13] Some occurrences of -ni- have been observed for medial immediate past or present tense. It is not clear if this is speaker dependent variation, or has significance.

[14] -e- is used with 1st person, and -a- with 2nd person.

[15] The sentence final indicative future tenses and subjects do not apply to the 2nd person. The subjunctive tense (-a-), and the subjunctive subjects (-a-, -ru-, -gu-) are used instead.

yare- and *-Ø-Ø-*. Otherwise the medial occurrence of *-Ø-yare-* would be incorrectly identified as encoding either immediate past or present tense (when it really only encodes present tense in the medial position) and the same for the medial occurrence of *-Ø-Ø-*. When the tense suffixes are combined with the subject suffixes, the only true ambiguity is *-Ø-na-*, which is ambiguous in that it encodes both sentence final 1st and 2nd person singular immediate past tense and sentence final 3rd person singular present tense.

1.2.3.5.10 Contrafactual Subject Suffixes

Contrafaction was described above in section 1.2.3.4.2 Contrafactual Exocenter. Below is a table of the Ogea contrafactual subject suffixes and some examples.

Table 19 Ogea Contrafactual Subject Suffixes

Person	**Number**	**Suffix**
1st	Singular	-e-
	Dual	-edere-
	Plural	-ede-
2nd, 3rd	All	-a-

```
(137) ji yafege
      ji yaf-e         -ge
      I  sit-Contra.S1s-Contra
      'if I had sat'

(138) ere  yafederege
      ere  yaf-edere     -ge
      we   sit-Contra.S1d-Contra
      'if we (dual) had sat'

(139) ere  yafedege
      ere  yaf-ede       -ge
      we   sit-Contra.S1p-Contra
      'if we (dual) had sat'

(140) yafage
      yaf-a               -ge
      sit-Contra.S2/3s/d/p-Contra
      'if you/he/she/they (sing/dual/plural) had sat'
```

1.2.3.5.11 Subjunctive Person and Number Suffixes

The Ogea subjunctive was described above in section 1.2.3.4.3 Subjunctive Exocenter. Below is a table of the subjunctive verb person suffixes, a table of the subjunctive verb number suffixes, and some examples. Note that subjunctive tenses and subjects do not distinguish between medial and final forms.

Table 20 Ogea Subjunctive Verb Person Suffixes

Person	**Suffix**
1st and 3rd	-o-
2nd	-a-

Table 21 Ogea Subjunctive Verb Number Suffixes

Person	**Number**	**Suffix**
1st	**All**	-u
2nd & 3rd	**Singular**	
	Dual	-ru
	Plural	-gu

The following examples of subjunctive suffixes were elicited from a native Ogea speaker. Note that number is undifferentiated in the first person, and that first person uses the singular form for singular, dual, and plural.

```
(141) ji   yafouganga
      ji   yaf-o      -u    -ga  -nga
      I    sit-Sj.S1/3-Sj.Ss-cert-int
      'I certainly will sit'

(142) ere yaforuganga
      ere yaf-o      -ru   -ga  -nga
      we  sit-Sj.S1/3-Sj.Sd-cert-int
      'We (dual) certainly will sit'

(143) ere  yafoguganga
      ere  yaf-o      -gu   -ga  -nga
      we   sit-Sj.S1/3-Sj.Sp-cert-int
      'We (plural) certainly will sit'
```

```
(144) ne   yafauganga
      ne   yaf-a    -u    -ga  -nga
      you  sit-Sj.S2-Sj.Ss-cert-int
      'you (sing) certainly will sit'

(145) tere yafaruganga
      tere yaf-a    -ru   -ga  -nga
      you  sit-Sj.S2-Sj.Sd-cert-int
      'you (dual) certainly will sit'

(146) tere yafaguganga
      tere yaf-a    -gu   -ga  -nga
      you  sit-Sj.S2-Sj.Sp-cert-int
      'you (plural) certainly will sit'

(147) no         yafouganga
      no         yaf-o    -u    -ga  -nga
      he/she/it  sit-Sj.S3-Sj.Ss-cert-int
      'he/she/it certainly will sit'

(148) nere yaforuganga
      nere yaf-o    -ru   -ga  -nga
      they sit-Sj.S3-Sj.Sd-cert-int
      'they(dual) certainly will sit'

(149) nere yafoguganga
      nere yaf-o    -gu   -ga  -nga
      they sit-Sj.S3-Sj.Sd-cert-int
      'they (plural) certainly will sit'
```

1.2.3.5.12 Infinitive Verb Number Suffixes

Infinitive verbs were discussed in section 1.2.3.4.4 Infinitive Exocenter. Below are a table and examples of infinitive verb number suffixes.

Table 22 Ogea Infinitive Verb Number Suffixes

Number	**Suffix**
Singular	-i
Dual	-ru
Plural	-gu

```
(150) isei             uuwa
      is   -e  -i      uuwa
      hear-Inf-Inf.Ss  not
      'I/you.sing/he/she did not hear'
```

```
(151) yaurenigei             nomo
      yaure-nig  -e  -i      nomo
      yell.to-O3p-Inf-Inf.Ss its
      'in order for I/you.sing/he/she to yell to them'

(152) umugeru         uuwa
      umug-e  -ru     uuwa
      fear-Inf-Inf.Sd not
      'we/you/they.dual were not afraid'

(153) yafegu         uuwa
      yaf-e  -gu     uuwa
      sit-Inf-Inf.Sp not
      'we/you/they.plural did not sit'

(154) haawewegu        nomo edo  uuwa
      haawe-we -gu     nomo edo  uuwa
      judge-Inf-Inf.Sp its  able not
      'we/you/they.plural are not able to judge'

(155) haawenigegu      nomo    boya
      haawe-nig-e  -gu nomo    bo   -ya
      judge-O3p-Inf-Inf.Sp its vAuxR-vAux.S1/2p
      'in order for we/you.plural to judge them'
```

1.2.3.6 Suppletive Verb Roots

Ogea has a limited number of suppletive verb roots that vary according to the object person and number. Suppletive forms exist for the verbs meaning 'to see', 'to hit', and 'to get'. It may be argued that a fourth set exists, namely for the verb 'to give', which is formed by using the object suffixes as verb roots. These four verb roots are the only known example of suppletion in Ogea. A paradigm is presented as a separate table for each suppletive verb (Table 23 through Table 26), and then together as a single matrix for comparative purposes in Table 27 . Note that there is a relationship between certain segments within each paradigm and across paradigms. This will be discussed in section **1.2.7 Templatic Morphology**, where a comparison will also be made to various pronouns and pronominal suffixes.

Table 23 Suppletive Verb Root Forms for 'See'

		Object Number		
		Singular	**Dual**	**Plural**
Object Person	**1st**	yeriye-	herire-	herige-
	2nd	neriye-	terire-	terige-
	3rd	ure-	nerire-	nerige-

Table 24 Suppletive Verb Root Forms for 'Hit'

		Object Number		
		Singular	**Dual**	**Plural**
Object Person	**1st**	yari-	harire-	harige-
	2nd	nari-	tarire-	tarige-
	3rd	wari-	narire-	narige-

Table 25 Suppletive Verb Root Forms for 'Get'

		Object Number		
		Singular	**Dual**	**Plural**
Object Person	**1st**	haife-	haire-	haige-
	2nd	haine-	hatire-	hatige-
	3rd	hau-	hanire-	hanige-

Table 26 Suppletive Verb Root Forms for 'Give'

		Object Number		
		Singular	**Dual**	**Plural**
Object Person	**1st**	hi-	hiri-	higi-
	2nd	hini-	tiri-	tigi-
	3rd	tu-	niri-	nigi-

Table 27 Comparative Matrix for All Suppletive Verb Roots

		Meaning	**Object Number**		
			Singular	**Dual**	**Plural**
Object Person	**1st**	see	yeriye-	herire-	herige-
		hit	yari-	harire-	harige-
		get	haife-	haire-	haige-
		give	hi-	hiri-	higi-
	2nd	see	neriye-	terire-	terige-
		hit	nari-	tarire-	tarige-
		get	haine-	hatire-	hatige-
		give	hini-	tiri-	tigi-
	3rd	see	ure-	nerire-	nerige-
		hit	wari-	narire-	narige-
		get	hau-	hanire-	hanige-
		give	tu-	niri-	nigi-

1.2.4 The Auxiliary Verb *bo-*

The auxiliary verb *bo-*[16] occurs both in sentence medial and sentence final positions. In the medial position it sometimes clearly encodes purpose, but in other cases it appears to act as a functor, encoding strictly grammatical rather than semantic information. In sentence final position, *bo-* semantically encodes what appears to be intent or purpose. Note that the exact meaning and function of *bo-* in a sentence final position is not known by this author and is difficult for Ogea speakers to explain. One Ogea speaker suggested that at one time Ogea differentiated immediate future and remote future, and that sentence final *bo-* encoded remote future. Note that *bo-* is suffixed for subject person and number. For all tenses, persons, and number except 2nd person future, the root *bo-* takes present tense indicative subject suffixes. In the examples below, *bo-* is

[16] A speaker dependent variant of *bo-* is *abo-*.

sometimes glossed by '?' when it is acting as a pure functor with no obvious semantic meaning.

Table 28 Subject Suffixes for the Root *bo-*

Number	Person	Tense	
		Non-Future	Future
Singular	1st	-ni	
	2nd	-ne	-nahe
	3rd	-na	
Dual	1st	-yare	
	2nd	-yare	-nadere
	3rd	-dere	
Plural	1st	-ya	
	2nd	-ya	-nade
	3rd	-de	

```
(156) no        wewe  waha               bona
      no        wewe  wa  -ha            bo   -na
      he/she/it dream that-pertaining.to vAuxR-vAux.S3s

      ninau        isinga   bona           yafai
      nina    -u   isi-nga  bo -na         yaf-a      -i
      thoughts-P3s hear-just vAuxR-vAux.S3s sit-Ind.Trp-Ind.S3s

      'He/she just sat thinking about that dream.'

(157) ji tere dada  etenga   boni
      ji tere dada  ete-nga  bo   -ni
      I  you  thing one-just vAuxR-vAux.S1s

      isokitigoni
      isoki-tig-o     -ni
      ask  -O2p-Ind.Tf-Ind.S1s

      'I will ask you about just one thing.'

(158) ji tere isaguganga                boni
      ji tere is  -a    -gu   -ga  -nga bo   -ni
      I  you  hear-Sj.S2-Sj.Ss-cert-int vAuxR-vAux.S1s

      gau    hina[17]
      gau    hi      -Ø           -na
      desire give.O1s-Ind.Tp-Ind.S3s
```

[17] In Ogea, various emotions are expressed as actions performed on a person and use a 3rd singular subject.

```
      'I desire that you will certainly listen!"
      Literally, 'Desire is given to me that you will
      certainly listen!'

(159) no      bagu goidu  bona
      no      bagu goi-du bo   -na
      he/she  with go -TS vAuxR-vAux.S3s

      letuwai
      le   -tu -wa     -i
      speak-O3s-Ind.Trp-Ind.S3s

      'He/she went to (him/her) and spoke to him/her.'

(160) isokinigona              bona
      isoki-nig-o     -na      bo   -na
      ask  -O3p-Ind.Tf-Ind.S3s vAuxR-vAux.S3s

      'In order to ask them…'

(161) onodu      bona           no
      ono    -du bo   -na,      no
      do.thus-SR vAuxR-vAux.S3s he/she/it

      nomo    bemu    umai
      nomo    bemu    um -a      -i
      his/her sibling die-Ind.Trp-Ind.S3s

      'Having done thus, his/her sibling died.'

(162) ne       haumu beha
      ne       haumu be  -ha
      you.sing place this-pertaining.to

      awadu    bonahe
      awa  -du bo   -nahe
      leave-TS vAuxR-vAux.Tf.S2s,

      ha    fere      langa  goyau
      ha    fere      la-nga goy-a    -u
      place different at-int go -Sj.S2-Sj.Ss

      'Leave this place and go to a different place!'

(163) ne       huwanyate   waha                bonahe
      ne       huwanya-te  wa  -ha             bo   -nahe
      you.sing insides-2sP that-pertaining.to vAuxR-vAux.Tf.S2s

      adai ou    hinou
      adai ou    hin     -ou
      not  heavy give.O2s-Nom

      'Do not be concerned about that!'
```

```
(164) tere       maidu   bonadere          uraru
      tere       mai -du bo   -nadere      ur     -a    -ru
      you.plural come-TS vAuxR-vAux.Tf.S2d see.O3s-Sj.S2-Sj.Sd

      'Come and see he/she/it!'

(165) tere            dada  beha               bonade
      tere            dada  be  -ha            bo   -nade
      you.dual/plural thing this-pertaining.to vAuxR-vAux.Tf.S2p

      ninatege     isagu
      nina-tege    is  -a    -gu
      thoughts-P2p hear-Sj.S2-Sj.Sp

      'Think about this thing!'
```

1.2.5 Nouns and Adjectives

1.2.5.1 Inalienably Possessed Nouns

In some Papuan languages (e.g., Hua [4]) all noun roots or stems may be suffixed with a possessive suffix. However, in Ogea the possessive suffixes attach only to inalienably possessed nouns such as body parts and kinship terms. Normally, even though an inalienably possessed noun is suffixed with a possessive suffix, the possessed noun will occur as part of a noun phrase, with the possessed noun preceded by a possessive pronoun as in (166).

```
(166) yame baife
      yame bai             -fe
      my   sibling.same.sex-P1s
      'my sibling.same.sex'

(167) nebere afenege
      nebere afe     -nege
      their  leg/foot-P3p
      'their legs'
```

The possessive suffixes are shown in Table 29.

Table 29 Ogea Possessive Suffixes

Person	Singular	Dual	Plural
1st	-fe	-re	-ge
2nd	-ne ~ -te[18]	-tere	-tege
3rd	-Ø or -u	-nere	-nege

1.2.5.2 Derived Nouns

The nominalizing suffix *-ou* is used to derive nouns from verbs in Ogea. The nominalized verbs are formed by replacing the exocenter with the nominalizing suffix, with the exception of the temporality suffix, which may be retained. This means that nominalized verbs may be fully inflected for object, benefaction, aspect, etc.

In Ogea, adjectives follow nouns. In some cases, the nominalized verb follows the noun, and appears to be functioning as an adjective. However, note that in the majority of cases, the nominalized verb precedes a noun, and thus fills the first noun slot of a compound noun phrase. In such cases the nominalized verb appears to be in an attributive relationship to the head noun. Also note that Ogea forms relative clauses by filling the head noun slot of a definite noun phrase, all the while retaining the full inflection of the exocenter. In example (168), note that (a) shows an independent clause, and (b) shows the same clause filling the head noun slot of a definite noun phrase, thus forming a restrictive relative clause. In the same example, (c) and (d) show the same verb nominalized with -ou. Note that in the data available to this author, for the nominalized form of 'die', (c) is the predominant pattern, with the pattern shown in (d)

[18] This is a speaker dependent variation. Most Ogea are at a minimum tri-lingual (Ogea, Tok Pisin, and one other language), and some speakers of Ogea have suggested the variation in the second singular possessive suffix is due to influence by the Uya language.

sometimes occurring. That is, 'umou' typically follows the head noun (as in (c)) rather than follows it (as in (d)). However, for the vast majority of examples of other nominalized verbs in the data (that is, verbs other than 'to die'), the nominalized verb precedes the head noun. Finally, note that relative clause formation as shown in (b) is very productive in Ogea, and is the basis for a variety of semantic relationships between clauses in Ogea (e.g., topicalization, conditions, hypotheticality).

```
(168) a. fai umai                 (an independent clause)
         fai um-a       -i
         man die-Ind.Trp-Ind.S3s
         '(A) man died.'

      b. fai umai               waha     (a relative clause)
         fai um -a      -i      wa  -ha
         man die-Ind.Trp-Ind.S3s that-pertaining.to
         'the man who died'

      c. fai umou       waha       (a nominalized verb following
         fai um -ou  wa  -ha       the head noun)
         man die-Nom that-pertaining.to
         'that dead man'

      d. umou    fai waha          (a nominalized verb preceding
         um -ou  fai wa  -ha        the head noun)
         die-Nom man that-pertaining.to
         'that dead man'

(169) a. fai no isina
         no     isi       -Ø     -na
         he/she understand-Ind.Tp-Ind.S3s
         'he/she understands'

      b. fai no isina                     waha
         fai no isi-Ø           -na       wa  -ha
         man he understand-Ind.Tp-Ind.S3s that-pertaining.to
         'the man who understands'

      c. isou              fai waha
         is-ou             fai wa  -ha
         understanding-Nom man that-pertaining.to
         'the understanding man'
      d. no     isou           hiloo bainga   bagu
         no     is        -ou  hiloo bai -nga bagu
         he/she understand-nom good  very-int with
         'He/she has very good understanding'

(170) hiiri katiwou      menekele
      hiiri kati    -wou menekele
      ocean encircle-Nom small
      'small enclosed ocean' (= a lake)

(171) ne nebere agotetenigigou       anyakaro
      ne nebere agotete-nigi-g  -ou  anyakaro
      you their teach  -O3p -Hab-Nom large
      'You are their leading teacher.'

(172) jirigou          mata
      jiri    -g  -ou  mata
      sleep.pl-Hab-Nom house
      'sleeping house' or 'bunkhouse'

(173) a. oofanigigai
         oofa      -nigi-g  -a      -i
         watch.over-O3p -Hab-Ind.Trp-Ind.S3s
         'he/she habitually watched over'

      b. oofanigigou        fai
         oofa      -nigi-g  -ou  fai
         watch.over-O3p -Hab-Nom man
         '(a) man (who) habitually watches over them'
          or 'a watchman'

(174) a. kurutuna
         kuru    -tu -Ø     -na
         look.for-O3p-Ind.Tp-Ind.S3s
         'he/she is looking for him/her'

      b. kurutuwou        haruwe
         kuru    -tu -wou haruwe
         look.for-O3s-Nom work
         '(the) work of looking for (something)'

(175) a. isiwai
         isi -wa     -i
         know-Ind.Trp-Ind.S3s
         'he/she knew'

      b. nomo    isou
         nomo    is  -ou
         his/her know-Nom
         'his/her knowledge'
```

```
(176) a. yeregai
         yereg-a      -i
         write-Ind.Trp-Ind.S3s
         'he/she wrote'

      b. nomo    yeregou
         nomo    yereg-ou
         his/her write-Nom
         'his/her writing'
```

The suffix *-ou* is also used with adjectives. The motivation for this is not understood by this author. Examples are shown below. In each of the examples, the same phrase would be grammatically correct if *-ou* were deleted. In each case, *-ou* seems to indicate that the noun has the quality of the adjective suffixed by *-ou*.

```
(177) ogola giiriwou
      ogola giiri-wou
      cloth red  -attribute
      'a red cloth'

(178) kesebu mutu kobibiwou
      kesebu mutu kobibi -wou
      knife  nose crooked-attribute
      'a knife with a crooked point'

(179) fai tiginiwou
      fai tigini  -wou
      man straight-attribute
      'a good (obedient) man'
```

Finally, note that a nominalized verb is ambiguous with the first and third singular subjunctive. For example, *lewou* may legitimately be parsed as either a nominalized noun ('speaker') or as a 1st or 3rd person subjunctive ('I/he/she/it will speak').

1.2.6 Pronouns

Ogea pronouns are not suffixed and therefore are not properly part of a morphological discussion of Ogea. However, because the Ogea pronouns are discussed in the section below on Ogea templatic morphology, tables are provided below for each type of Ogea pronoun. It should be noted that Ogea pronouns do not encode gender.

1.2.6.1 Personal Pronouns

The same forms of Ogea personal pronouns are used to encode both grammatical subjects and objects. Translations include *I, me, you, he, she, it, they,* and *them.*

Table 30 Ogea Personal Pronouns

		Number	
		Singular	**Dual/Plural**
Person	**1st**	ji	ere
	2nd	ne	tere
	3rd	no	nere

1.2.6.2 Intensive/Reflexive Pronouns

The intensive/reflexive pronouns are translated into English as *myself, yourself, himself, herself, itself,* and *themselves*.

Table 31 Ogea Intensive/Reflexive Pronouns

		Number		
		Singular	**Dual**	**Plural**
Person	**1st**	yage	hare	hage
	2nd	nage	tare	tage
	3rd	nogo	nare	nage

1.2.6.3 Possessive Pronouns

The possessive pronouns are translated into English as *my, your, his, her, its,* and *their*.

Table 32 Ogea Possessive Pronouns

		Number		
		Singular	**Dual**	**Plural**
Person	**1st**	yame	heire	ebere
	2nd	name	teire	tebere
	3rd	nomo	neire	nebere

1.2.7 Templatic Morphology

It was noted in [1] that a relationship exists between various forms of the Ogea suppletive verbs. That such a relationship exists between not only the suppletive verbs but also the pronouns and a number of suffix types was not formally noted. Furthermore, no attempt was made to describe these relationships. This section, therefore, attempts to describe the relationships. While the description of such relationships likely has no pragmatic use in the natural language processing of Ogea, such a description is of interest for historical (diachronic) and comparative purposes. That is, the relationships may prove useful in studying the history of the Ogea language, and are certainly useful in comparing Ogea to other related languages such as Siroi (as will be discussed below). Also, these relationships are used elsewhere in this dissertation to illustrate the ability of DATR to capture linguistic generalizations not possible to capture in AMPLE's LKR.

The approach used in the analysis presented here is a templatic one. For convenience, the various tables presented above for suppletive verb roots, various suffixes, and the pronouns are presented as a single table below (Table 33). Examination of the forms for 1st, 2nd, and 3rd person in the dual and plural columns shows an interesting phenomenon. A basic template is evident that has the shape CV(CV)CV, where the initial consonant is a segment

Table 33 Various Ogea Verb Roots, Suffixes, and Pronouns

		Meaning	Number		
			Singular	Dual	Plural
Person	1st	see	yeriye-	**h**eri**r**e-	**h**eri**g**e-
		hit	yari-	**h**ari**r**e-	**h**ari**g**e-
		get	haife-	**ha** i**r**e-	**h** ai**g**e-
		give	hi-	**h** i**r**i-	**h** i**g**i-
		Object	-hi-	-**h** i**r**i-	-h i**g**i-
		Benefactive	-hai-	-**h** a**r**i-	-**h** a**g**i-
		Possessive Suffix	-fe	-**r**e	-**g**e
		Reflexive Pronoun	yage	**h** a**r**e	**h** a**g**e
		Possessive Pronoun	yame	**h** ei**r**e	ebe**r**e
		Personal Pronoun	ji	e**r**e	e**r**e
	2nd	see	neriye-	**t**eri**r**e-	**t**eri**g**e-
		hit	nari-	**t**ari**r**e-	**t**ari**g**e-
		get	haine-	ha**t** i**r**e-	ha**t** i**g**e-
		give	hini-	**t** i**r**i-	**t** i**g**i-
		Object	-hini-	-**t** i**r**i-	-**t** i**g**i-
		Benefactive	-hani-	-**t** a**r**i-	-**t** a**g**i-
		Possessive suffix	-ne	-**t** e**r**e	-**t** e**g**e
		Reflexive Pronoun	nage	**t** a**r**e	**t** a**g**e
		Possessive Pronoun	name	**te** i**r**e	**t**ebe**r**e
		Personal Pronoun	ne	**t** e**r**e	**t** e**r**e
	3rd	see	ure-	**n**eri**r**e-	**n**eri**g**e-
		hit	wari-	**n**ari**r**e-	**n**ari**g**e-
		get	hau-	ha**n** i**r**e-	ha**n** i**g**e-
		give	tu-	**n** i**r**i-	**n** i**g**i-
		Object	-tu-	-**n** i**r**i-	-**n** i**g**i-
		Benefactive	-fu-	-**n** a**r**i-	-**n** a**g**i-
		Possessive Suffix	-mu, -0	-**n** e**r**e	-**n** e**g**e
		Reflexive Pronoun	nogo	**n** a**r**e	**n** a**g**e
		Possessive Pronoun	nomo	**n** ei**r**e	**n**ebe**r**e
		Personal Pronoun	no	**n** e**r**e	**n** e**r**e

that indicates number (dual and plural), the middle V(CV) encodes the core meaning, and the final consonant is a segment that indicates person (1st, 2nd, 3rd). While the current Ogea surface forms contain exceptions to the postulated template, historically the forms may have been regular for all lexemes involved.

Basic generalizations about the dual and plural forms will now be made, followed by a discussion of the exceptions. First, the initial consonant is either /h/, /t/, or /n/, with

/h/ indicating 1st person, /t/ indicating 2nd person, and /n/ indicating 3rd person. Note that it is possible to analyze the initial consonant as a single segment prefix indicating person. However, given that Ogea is an otherwise suffixing language, and that the person segment occurs as the initial segment of certain suffixes, it seems more productive to this author to use a templatic approach to the analysis rather than postulate a prefix that attaches to certain suffixes. The second generalization that may be made is that the final consonant indicates number, with /r/ indicating dual and /g/ indicating plural. However, if the lexeme does not distinguish between dual and plural, /r/ is used to encode both dual and plural (e.g., see the entries for the plural possessive pronouns and personal pronouns). The third observation is that the final vowel is /i/ for morphemes that are suffixes, and /e/ for morphemes that are verb roots, pronouns, or noun suffixes. Thus, the underlying template is {h,t,n}V(CV){r,g}V. This template is presented as a paradigm based on person and number in Table 34.

Table 34 Paradigmatic Presentation of Person/Number Template

		Number	
		Dual	**Plural**
Person	**First**	**h**V(CV)**r**V	**h**V(CV)**g**V
	Second	**t**V(CV)**r**V	**t**V(CV)**g**V
	Third	**n**V(CV)**r**V	**n**V(CV)**g**V

Now discussion will turn to exceptions to the basic template. The exceptions include absence of the initial person consonant segment, the presence of a CV before the initial person consonant segment, the absence of an initial CV, and the use of /r/ to indicate both dual and plural.

The initial person consonant segment is missing in the 1st person dual and plural forms for the possessive suffix and the personal pronoun, and in the 1st person plural form of the possessive pronoun. It is, of course, entirely possible that historically the initial /h/ was present, but has been lost. Historically the 1st person dual and plural pronoun may have been /here/.

In the case of the suppletive verb root for 'get' (a person), in the 2nd and 3rd person dual and plural forms, /ha/ precedes the person consonant segment. The segmentation of the verb for 'get' is problematic for the 1st person dual and plural. Historically was the dual form /hahaire/ and the plural form /hahaige/? Was the initial /ha/ lost, or the second?

Regarding the 1st person possessive suffix, the problem is not the presence of a syllable before the person segment, but rather the opposite. No initial syllable occurs before the one indicating number. That is, rather than /-hare/ for dual, and /-hage/ for plural, the forms are /-re/ and /-ge/.

The last exception to the overall template is seen in the forms for the possessive and personal pronoun for all persons in the dual and plural. Rather than the expected /g/ segment occurring as the final consonant, /r/ occurs. When the distinction is only between singular and plural rather than between singular, dual, and plural, the /r/ is used to encode both dual and plural.

A number of interesting questions arise from the data presented here, the answers to which are not known to this author. First, why is it that the first person forms appear to lack the regularity exhibited by the dual and plural forms in terms of a consonant segment encoding person, and a consonant segment encoding number? Second, why do the suppletive verb roots exist at all? They are the exception to an overwhelming majority of verb roots that merely encode the basic verb meaning, and leave the encoding of object person and number to separate suffixes. Are the suppletive verb roots the result of borrowing from other languages? If they were borrowed, why do they have a systematic relationship to the pronouns and possessive suffix? Or, historically did these particular verbs (see, hit, get, give) encode actions considered of more importance than others? Third, do languages related to Ogea also exhibit such a template as the one described above? This may be the case.

As for the third question, this author has access to data for only one closely related language, namely Siroi, described by Wells [7]. This data indicates that at least one related language has such a template. Wells provides paradigms for the Siroi personal and possessive pronouns and the object suffixes (combined into a single table below), but does not provide a paradigm for benefactive suffixes or reflexive pronouns[19]. Also, there is no mention made of suppletive verb roots that vary according to object person in Siroi.

Table 35 Siroi Object Suffixes, Possessive, and Personal Pronouns

		Meaning	**Number**		
			Singular	**Dual**	**Plural**
Person	**1st**	Object	-y-	-**s**ik-	-**s**ing-
		Possessive Pronoun	yiŋe	**s**ikile	**s**ingine
		Personal Pronoun	ye	**s** ile	**s** ine
	2nd	Object	-n-	-**t**ik-	-**t**ing-
		Possessive Pronoun	naŋe	**t**akile	**t**angine
		Personal Pronoun	ne	**t** ale	**t** ane
	3rd	Object	-Ø-	-**n**ik-	-**n**ing-
		Possessive Pronoun	nuŋe	**n**akile	**n**angine
		Personal Pronoun	nu	**n** ale	**n** ane

[19] Per Sjaak and Jacqueline van Kleef (SIL), personal communication, Siroi does not have benefactive suffixes or reflexive pronouns.

In examining Table 35, as with Ogea, in Siroi a pattern may be observed in the dual and plural forms. However, in the case of Siroi, although there is a clear initial segment indicating person (/s/ for 1st, /t/ for 2nd, and /n/ for 3rd), there is not as clear a pattern for a final number consonant segment as there is with Ogea. Either /k/ or /l/ or both may occur for duals, and /ng/ or /n/ or both may occur for plurals in the Siroi data. It is also interesting to note that both Ogea and Siroi share the same segments for the 2nd and 3rd person forms. Both use /t/ to indicate 2nd person, and both use /n/ to indicate 3rd person.

1.3 Summary

This appendix has described Ogea morphophonemics and morphotactics at a level sufficient for the reader to understand discussion of the AMPLE-based and DATR-based lexicons found in other appendices. In addition, this appendix has discussed the relationship between various suppletive verbs, pronouns, and possessive suffixes, and hypothesized an underlying template to describe these relationships.

1.4 References

[1] M. Colburn, “Erima Grammar Essentials,” Summer Institute of Linguistics, Unpublished Manuscript. Ukarumpa, Papua New Guinea 1979.

[2] M. Colburn, “The Functions and Meanings of the Erima Deictic Articles,” *Pacific Linguistics A-69, Papers in New Guinea Linguistics*, vol. 23, pp. 209-272, 1984.

[3] B. F. Grimes and J. E. Grimes, “Ethnologue: Language Family Index,” . Dallas, TX: Summer Institute of Linguistics, 1997.

[4] J. Haiman, “Hua (Papuan),” in *The Handbook of Morphology*, A. Spencer and A. M. Zwicky, Eds. Oxford: Blackwell Publishers, Ltd., 1998, pp. 539-562.

[5] N. N. Mikloucho-Maclay, *Mikloucho-Maclay: New Guinea Diaries 1871-1883*. Madang, Papua New Guinea: Kristen Press, 1975.

[6] P. Staalesen, *Clause Relationships in Iatmul*, vol. A-31:45-69. Canberra: Australian National University, 1972.

[7] M. A. Wells, *Siroi Grammar*, vol. B-51. Canberra: Australian National University, 1979.

APPENDIX 2

OGEA AMPLE LEXICON

A2.1 Description

This appendix provides a listing of the Ogea AMPLE lexicon file *Ogea.db*. The database file uses a "unified" dictionary, with each record type identifying the entry as either a root ('r') or suffix ('s'). The unique identifier for each record is the morpheme name (\g). In addition to the type field and the morpheme name field, five additional AMPLE field codes were required to properly parse Ogea data. The field codes have the following meaning:

Code	**Meaning**
\a	allomorph
\c	category
\g	morphname
\ge	English gloss
\mp	morpheme property
\t	type
\u	underlying form

Note that in addition to lexicon (dictionary) database files, AMPLE uses other files, including control files. These files are not presented here since they would not be maintained and generated from DATR. The records entries listed provide the data required to parse all examples found in Appendix 1.

A2.2 The Listing

Filename: ogea.db.

Note: line numbers added for reference only. Not in actual file.

```
(180) \g able
(181) \ge able
(182) \type r
(183) \a edo
(184) \u edo
(185) \c Adv
(186)
(187) \g allow/leave
(188) \ge allow/leave
(189) \type r
(190) \a awa
(191) \a aw / _ [V]
(192) \u awa
(193) \mp MC1
(194) \c VR
(195)
(196) \g always
(197) \ge always
(198) \type r
(199) \a gaigai
(200) \u gaigai
(201) \c Adv
(202)
(203) \g animal
(204) \ge animal
(205) \type r
(206) \a kaafa
(207) \u kaafa
(208) \c NW
(209)
(210) \g appear/arrive
(211) \ge appear/arrive
(212) \type r
(213) \a heige
(214) \a heig / _ [V]
(215) \u heige
(216) \mp MC1
(217) \c VR
(218)
(219) \g ask
(220) \ge ask
(221) \type r
(222) \a isoki
(223) \u isoki
(224) \mp MC2
(225) \c VR
(226)
(227) \g at
(228) \ge at
(229) \type r
(230) \a la
(231) \u la
(232) \c Prep
(233)
(234) \g Attrib
(235) \ge Attributive marker
(236) \type s
```

```
(237) \a ou / [C] _
(238) \a wou / [V] _
(239) \u ou
(240) \o 110
(241) \c Adj/Atr
(242)
(243) \g B1d
(244) \ge for us (dual)
(245) \type s
(246) \a hari
(247) \a har / _ [V]
(248) \u hari
(249) \mp MC1
(250) \o 60
(251) \c VR/VR
(252)
(253) \g B1p
(254) \ge for us (plural)
(255) \type s
(256) \a hagi
(257) \a hag / _ [V]
(258) \u hagi
(259) \mp MC1
(260) \o 60
(261) \c VR/VR
(262)
(263) \g B1s
(264) \ge for me
(265) \type s
(266) \a hai
(267) \a hay / _ [V]
(268) \u hai
(269) \mp MC2
(270) \o 60
(271) \c VR/VR
(272)
(273) \g B2d
(274) \ge for you (dual)
(275) \type s
(276) \a tari
(277) \a tar / _ [V]
(278) \u tari
(279) \mp MC1
(280) \o 60
(281) \c VR/VR
(282)
(283) \g B2p
(284) \ge for you (plural)
(285) \type s
(286) \a tagi
(287) \a tag / _ [V]
(288) \u tagi
(289) \mp MC1
(290) \o 60
(291) \c VR/VR
(292)
(293) \g B2s
```

```
\ge for you (singular)
\type s
\a hani
\a han / _ [V]
\u hani
\mp MC1
\o 60
\c VR/VR

\g B3d
\ge for them (dual)
\type s
\a nari
\a nar / _ [V]
\u nari
\mp MC1
\o 60
\c VR/VR

\g B3p
\ge for them (plural)
\type s
\a nagi
\a nag / _ [V]
\u nagi
\mp MC1
\o 60
\c VR/VR

\g B3s
\ge for him/her/it
\type s
\a fu
\u fu
\mp MC2
\o 60
\c VR/VR

\g bad
\ge bad
\type r
\a inyaba
\u inyaba
\c Adj

\g be
\ge be
\type r
\a inyi
\a iny / _ [V]
\u inyi
\mp MC1
\c VR

\g become.bad
\ge become.bad
\type r
```

```
(351) \a inyaba +/ ~_ TO
(352) \u inyaba_2
(353) \mp MC2
(354) \c VR
(355)
(356) \g become.crooked
(357) \ge become.crooked
(358) \type r
(359) \a kobibi +/ ~_ TO
(360) \u kobibi_2
(361) \mp MC2
(362) \c VR
(363)
(364) \g become.good
(365) \ge become.good
(366) \type r
(367) \a hilou +/ ~_ TO
(368) \a hiloo +/ ~_ TO
(369) \u hilou_2
(370) \mp MC2
(371) \c VR
(372)
(373) \g become.heavy
(374) \ge become.heavy
(375) \type r
(376) \a ou +/ ~_ TO
(377) \u ou_2
(378) \mp MC2
(379) \c VR
(380)
(381) \g become.large
(382) \ge become.large
(383) \type r
(384) \a anyakaro +/ ~_ TO
(385) \u anyakaro_2
(386) \mp MC2
(387) \c VR
(388)
(389) \g become.many
(390) \ge become.many
(391) \type r
(392) \a baingaro +/ ~_ TO
(393) \u baingaro_2
(394) \mp MC2
(395) \c VR
(396)
(397) \g become.nothing
(398) \ge become.nothing
(399) \type r
(400) \a uuwa +/ ~_ TO
(401) \u uuwa_2
(402) \mp MC2
(403) \c VR
(404)
(405) \g become.red
(406) \ge become.red
(407) \type r
```

```
(408) \a giiri +/ ~_ TO
(409) \u giiri_2
(410) \mp MC2
(411) \c VR
(412)
(413) \g become.short
(414) \ge become.short
(415) \type r
(416) \a tutu +/ ~_ TO
(417) \u tutu_2
(418) \mp MC2
(419) \c VR
(420)
(421) \g become.small
(422) \ge become.small
(423) \type r
(424) \a menekele +/ ~_ TO
(425) \a melekene +/ ~_ TO
(426) \u menekele_2
(427) \mp MC2
(428) \c VR
(429)
(430) \g become.straight
(431) \ge become.straight
(432) \type r
(433) \a tigini +/ ~_ TO
(434) \u tigini_2
(435) \mp MC2
(436) \c VR
(437)
(438) \g become.true
(439) \ge become.true
(440) \type r
(441) \a ngalenga +/ ~_ TO
(442) \u ngalenga_2
(443) \mp MC2
(444) \c VR
(445)
(446) \g become.wild
(447) \ge become.wild
(448) \type r
(449) \a kui +/ ~_ TO
(450) \u kui_2
(451) \mp MC2
(452) \c VR
(453)
(454) \g bound
(455) \ge bound
(456) \type r
(457) \a dage
(458) \a dag / _ [V]
(459) \u dage
(460) \mp MC1
(461) \c VR
(462)
(463) \g break.in.two
(464) \ge break.in.two
```

```
(465) \type r
(466) \a kou
(467) \a ko
(468) \u kou
(469) \mp MC2 RC1
(470) \c VR
(471)
(472) \g causative
(473) \ge causative
(474) \type s
(475) \a mu
(476) \u mu
(477) \o 10
(478) \c VR/VR
(479)
(480) \g certainty
(481) \ge certainty
(482) \type s
(483) \a nga
(484) \u nga
(485) \o 110
(486) \c VSjS/VSjS
(487)
(488) \g chase
(489) \ge chase
(490) \type r
(491) \a soo
(492) \u soo
(493) \mp MC2
(494) \c VR
(495)
(496) \g cloth
(497) \ge cloth
(498) \type r
(499) \a ogola
(500) \u ogola
(501) \c NW
(502)
(503) \g come.down
(504) \ge come.down
(505) \type r
(506) \a me
(507) \a m / _ [V]
(508) \u me
(509) \mp MC1
(510) \c VR
(511)
(512) \g come.up
(513) \ge come.up
(514) \type r
(515) \a mai
(516) \a may / _ [V]
(517) \u mai
(518) \mp MC2
(519) \c VR
(520)
(521) \g completely
```

```
\ge completely
\type s
\a boro
\u boro
\c VR/VR

\g connect/join
\ge connect/join
\type r
\a bagumu
\u bagu-mu
\mp MC2
\c VR

\g consume
\ge consume
\type r
\a nyi
\a ny / _ [V]
\u nyi
\mp MC1
\c VR

\g Contra
\ge Contrafactual
\type s
\a ge +/ [Contra.Subj] _
\u ge
\o 100
\c Contra/Contra

\g Contra.S1d
\ge Contractual first dual subject
\type s
\a edere / [C] _ +/ _ Contra
\a wedere / [V] _ +/ _ Contra
\u edere
\o 80
\c VR/Contra SMV/Contra

\g Contra.S1p
\ge Contractual first plural subject
\type s
\a ede / [C] _ +/ _ Contra
\a wede / [V] _ +/ _ Contra
\u ede
\o 80
\c VR/Contra SMV/Contra

\g Contra.S1s
\ge Contractual first singular subject
\type s
\a e / [C] _ +/ _ Contra
\a we / [V] _ +/ _ Contra
\u e
\o 80
\c VR/Contra SMV/Contra
```

```

\g Contra.S2/3s/d/p
\ge Contractual second or third singular, dual,
                     or plural subject
\type s
\a a / [C] _ +/ _ Contra
\a wa / [V] _ +/ _ Contra
\u a
\o 80
\c VR/Contra SMV/Contra

\g crooked
\ge crooked
\type r
\a kobibi
\u kobibi
\c Adj

\g cut
\ge cut
\type r
\a karu
\a ka
\u karu
\mp MC2 RC1
\c VR

\g deflect
\ge deflect
\type r
\a ariri
\u ariri
\mp MC2
\c VR

\g desire
\ge desire
\type r
\a gau
\u gau
\c NW

\g die
\ge die
\type r
\a ume
\a um / _ [V]
\u ume
\mp MC1
\c VR

\g different
\ge different
\type r
\a fere
\u fere
\c Adj
```

```
(636)
(637) \g do.not
(638) \ge do.not
(639) \type r
(640) \a adai
(641) \u adai
(642) \c Adv
(643)
(644) \g do.thus
(645) \ge do.thus
(646) \type r
(647) \a ono
(648) \u ono
(649) \mp MC2
(650) \c VR
(651)
(652) \g do.work
(653) \ge do.work
(654) \type r
(655) \a haruwe +/ ~_ TO
(656) \u haruwe_2
(657) \mp MC2
(658) \c VR
(659)
(660) \g dream
(661) \ge dream
(662) \type r
(663) \a wewe
(664) \u wewe
(665) \c NW
(666)
(667) \g encircle
(668) \ge encircle
(669) \type r
(670) \a kati
(671) \u kati
(672) \mp MC2
(673) \c VR
(674)
(675) \g face
(676) \ge face
(677) \type r
(678) \a hogo
(679) \u hogo
(680) \c PNR
(681)
(682) \g fear
(683) \ge fear
(684) \type r
(685) \a umuge
(686) \a umug / _ [V]
(687) \u umuge
(688) \mp MC1
(689) \c VR
(690)
(691) \g fight
(692) \ge fight
```

```
\type r
\a wara
\a war / _ [V]
\u wara
\mp MC1
\c VR

\g fill
\ge fill
\type r
\a bolo
\u bolo
\mp MC2
\c VR

\g finish
\ge finish
\type r
\a usu
\u usu
\mp MC2
\c VR

\g fly
\ge fly
\type r
\a fililai
\u fililai
\mp MC2
\c VR

\g follow
\ge follow
\type r
\a oojo
\u oojo
\mp MC2
\c VR

\g gather
\ge gather
\type r
\a elege
\a eleg / _ [V]
\u elege
\mp MC1
\c VR

\g get.O1d
\ge get us (dual)
\type r
\a haire
\a hair / _ [V]
\u haire
\mp MC1
\c VR

```

```
\g get.O1p
\ge get us (plural)
\type r
\a haige
\a haig / _ [V]
\u haige
\mp MC1
\c VR

\g get.O1s
\ge get me
\type r
\a haife
\a haif / _ [V]
\u haife
\mp MC1
\c VR

\g get.O2d
\ge get you (dual)
\type r
\a hatire
\a hatir / _ [V]
\u hatire
\mp MC1
\c VR

\g get.O2p
\ge get you (plural)
\type r
\a hatige
\a hatig / _ [V]
\u hatige
\mp MC1
\c VR

\g get.O2s
\ge get you (singular)
\type r
\a haine
\a hain / _ [V]
\u haine
\mp MC1
\c VR

\g get.O3d
\ge get them (dual)
\type r
\a hanire
\a hanir / _ [V]
\u hanire
\mp MC1
\c VR

\g get.O3p
\ge get them (plural)
\type r
```

```
(807) \a hanige
(808) \a hanig / _ [V]
(809) \u hanige
(810) \mp MC1
(811) \c VR
(812)
(813) \g get.O3s
(814) \ge get him/her/it
(815) \type r
(816) \a hau
(817) \u hau
(818) \mp MC2
(819) \c VR
(820)
(821) \g get.up
(822) \ge get.up
(823) \type r
(824) \a jai
(825) \a jay / _ [V]
(826) \u jai
(827) \mp MC2
(828) \c VR
(829)
(830) \g get/take
(831) \ge get/take
(832) \type r
(833) \a te
(834) \a t / _ [V]
(835) \u te
(836) \mp MC1
(837) \c VR
(838)
(839) \g give.O1d
(840) \ge give us (dual)
(841) \type r
(842) \a hiri
(843) \a hir / _ [V]
(844) \u hiri
(845) \mp MC1
(846) \c VR
(847)
(848) \g give.O1p
(849) \ge give us (plural)
(850) \type r
(851) \a higi
(852) \a hig / _ [V]
(853) \u higi
(854) \mp MC1
(855) \c VR
(856)
(857) \g give.O1s
(858) \ge give me
(859) \type r
(860) \a hii
(861) \a hiy / _ [V]
(862) \u hii
(863) \mp MC2
```

```
\c VR

\g give.O2d
\ge give you (dual)
\type r
\a tiri
\a tir / _ [V]
\u tiri
\mp MC1
\c VR

\g give.O2p
\ge give you (plural)
\type r
\a tigi
\a tig / _ [V]
\u tigi
\mp MC1
\c VR

\g give.O2s
\ge give you (singular)
\type r
\a hini
\a hin / _ [V]
\u hini
\mp MC1
\c VR

\g give.O3d
\ge give them (dual)
\type r
\a niri
\a nir / _ [V]
\u niri
\mp MC1
\c VR

\g give.O3p
\ge give them (plural)
\type r
\a nigi
\a nig / _ [V]
\u nigi
\mp MC1
\c VR

\g give.O3s
\ge give him/her/it
\type r
\a tu
\u tu
\mp MC2
\c VR

\g go
\ge go
```

```
(921) \type r
(922) \a goi
(923) \a goy / _ [V]
(924) \u goi
(925) \mp MC2
(926) \c VR
(927)
(928) \g go.down
(929) \ge go.down
(930) \type r
(931) \a mini
(932) \a min / _ [V]
(933) \u mini
(934) \mp MC1
(935) \c VR
(936)
(937) \g good
(938) \ge good
(939) \type r
(940) \a hilou
(941) \a hiloo
(942) \u hilou
(943) \c Adj
(944)
(945) \g Hab
(946) \ge Habitual
(947) \type s
(948) \a g
(949) \u g
(950) \o 70
(951) \c VR/SFV VR/Nom
(952)
(953) \g hand
(954) \ge hand
(955) \type r
(956) \a owo
(957) \u owo
(958) \c PNR
(959)
(960) \g he/she/it
(961) \ge he/she/it
(962) \type r
(963) \a no
(964) \u no
(965) \c PerPro
(966)
(967) \g hear/understand/obey
(968) \ge hear/understand/obey
(969) \type r
(970) \a isi
(971) \a is / _ [V]
(972) \u isi
(973) \mp MC1
(974) \c VR
(975)
(976) \g heavy
(977) \ge heavy
```

```
(978) \type r
(979) \a ou
(980) \u ou
(981) \c Adj
(982)
(983) \g help
(984) \ge help
(985) \type r
(986) \a hoyo
(987) \u hoyo
(988) \mp MC2
(989) \c VR
(990)
(991) \g himself/herself/itself
(992) \ge himself/herself/itself
(993) \type r
(994) \a nogo
(995) \u nogo
(996) \c RflPro
(997)
(998) \g his/her/its
(999) \ge his/her/its
(1000)      \type r
(1001)      \a nomo
(1002)      \u nomo
(1003)      \c PosPro
(1004)
(1005)      \g hit.O1d
(1006)      \ge hit us (dual)
(1007)      \type r
(1008)      \a harire
(1009)      \a harir / _ [V]
(1010)      \u harire
(1011)      \mp MC1
(1012)      \c VR
(1013)
(1014)      \g hit.O1p
(1015)      \ge hit us (plural)
(1016)      \type r
(1017)      \a harige
(1018)      \a harig / _ [V]
(1019)      \u harige
(1020)      \mp MC1
(1021)      \c VR
(1022)
(1023)      \g hit.O1s
(1024)      \ge hit me
(1025)      \type r
(1026)      \a yari
(1027)      \a yar / _ [V]
(1028)      \u yari
(1029)      \mp MC1
(1030)      \c VR
(1031)
(1032)      \g hit.O2d
(1033)      \ge hit you (dual)
(1034)      \type r
```

```
\a tarire
\a tarir / _ [V]
\u tarire
\mp MC1
\c VR

\g hit.O2p
\ge hit you (plural)
\type r
\a tarige
\a tarig / _ [V]
\u tarige
\mp MC1
\c VR

\g hit.O2s
\ge hit you (singular)
\type r
\a nari
\a nar / _ [V]
\u nari
\mp MC1
\c VR

\g hit.O3d
\ge hit them (dual)
\type r
\a narire
\a narir / _ [V]
\u narire
\mp MC1
\c VR

\g hit.O3p
\ge hit them (plural)
\type r
\a narige
\a narig / _ [V]
\u narige
\mp MC1
\c VR

\g hit.O3s
\ge hit him/her/it
\type r
\a wari
\a war / _ [V]
\u wari
\mp MC1
\c VR

\g I
\ge I
\type r
\a ji
\u ji
\c PerPro
```

```

\g idle
\ge idle
\type r
\a ewe
\u ewe
\c Adj

\g Ind.S1d
\ge Indicative first dual subject
\type s
\a 0 / o _ # +/ Ind.Tf _ ~SR
\a dere / e _ +/ Ind.Tf _ SR
\a re / e _ +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\a yare +/ [Ind.Tip.Tp] _
\a yare / o _ +/ Ind.Tf _ ~SR
\u yare
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV

\g Ind.S1p
\ge Indicative first plural subject
\type s
\a 0 / o _ # +/ Ind.Tf _ ~SR
\a de / e _ +/ Ind.Tf _ SR
\a ya +/ [Ind.Tip.Tp] _
\a ya / o _ +/ Ind.Tf _ ~SR
\a ye / e _ +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\u ya
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV

\g Ind.S1s
\ge Indicative first singular subject
\type s
\a he / e _ +/ Ind.Tf _ SR
\a n / o _ # +/ Ind.Tf _ ~SR
\a na +/ Ind.Tip _
\a ne +/ [Ind.Tip.Tp] _ SR
\a ne / e _ +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\a ni +/ Ind.Tp _ ~SR +/ [Ind.Tip.Tp] _ SR
\a ni / o _ +/ Ind.Tf _ ~SR
\u ni
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV

\g Ind.S2d
\ge Indicative second dual subject
\type s
\a dere / a _ +/ Ind.Tf _ SR
\a re / e _ +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\a yare +/ [Ind.Tip.Tp] _
\u yare
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV

\g Ind.S2p
```

```
(1149)      \ge Indicative second plural subject
(1150)      \type s
(1151)      \a de / a _ +/ Ind.Tf _ SR
(1152)      \a ya +/ [Ind.Tip.Tp] _
(1153)      \a ye / e _ +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
(1154)      \u ya
(1155)      \o 90
(1156)      \c SFV/SFV SFV/SFVW SMV/SMV
(1157)
(1158)      \g Ind.S2s
(1159)      \ge Indicative second singular subject
(1160)      \type s
(1161)      \a he / a _ +/ Ind.Tf _ SR
(1162)      \a na +/ Ind.Tip _
(1163)      \a ne +/ Ind.Tp _ SR
(1164)      \a ne / e _ +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
(1165)      \a ni +/ Ind.Tp _ ~SR +/ [Ind.Tip.Tp] _ SR
(1166)      \u ni
(1167)      \o 90
(1168)      \c SFV/SFV SFV/SFVW SMV/SMV
(1169)
(1170)      \g Ind.S3d
(1171)      \ge Indicative third dual subject
(1172)      \type s
(1173)      \a d / o _ # +/ Ind.Tf _ ~SR
(1174)      \a dere +/ Ind.Tp _ ~SR
(1175)      \a dere / o _ +/ Ind.Tf _ ~SR
(1176)      \a re / a _ +/ Ind.Tip _ +/ Ind.Tp _ SR +/ Ind.Trp _
(1177)      \a ro / o _ +/ Ind.Tf _ SR
(1178)      \u dere
(1179)      \o 90
(1180)      \c SFV/SFV SFV/SFVW SMV/SMV
(1181)
(1182)      \g Ind.S3p
(1183)      \ge Indicative third plural subject
(1184)      \type s
(1185)      \a d / o _ # +/ Ind.Tf _ ~SR
(1186)      \a de +/ Ind.Tp _ ~SR
(1187)      \a de / o _ +/ Ind.Tf _
(1188)      \a go / o _ +/ Ind.Tf _ SR
(1189)      \a ne / a _ +/ Ind.Tip _ +/ Ind.Tp _ SR +/ Ind.Trp _
(1190)      \u de
(1191)      \o 90
(1192)      \c SFV/SFV SFV/SFVW SMV/SMV
(1193)
(1194)      \g Ind.S3s
(1195)      \ge Indicative third singular subject
(1196)      \type s
(1197)      \a 0 / a _ +/ Ind.Tip _ +/ Ind.Tp _ SR
(1198)      \a 0 / o _ +/ Ind.Tf _ SR
(1199)      \a i / a _ +/ Ind.Trp _
(1200)      \a n / o _ # +/ Ind.Tf _ ~SR
(1201)      \a na +/ Ind.Tp _ ~SR
(1202)      \a ta +/ Ind.Tip _ ~SR
(1203)      \a na / o _ +/ Ind.Tf _ ~SR
(1204)      \u na
(1205)      \o 90
```

```
\c SFV/SFV SFV/SFVW SMV/SMV

\g Ind.Tf
\ge Indicative future tense
\type s
\a e / [C] _ +/ _ [Ind.S1]
\a we / [V] _ +/ _ [Ind.S1]
\a a / [C] _ +/ _ [Ind.S2]
\a wa / [V] _ +/ _ [Ind.S2]
\a o / [C] _ +/ _ [Ind.S1.S3]
\a wo / [V] _ +/ _ [Ind.S1.S3]
\u o
\o 80
\c SMV/SMV VR/SMV VR/SFV SFV/SFV

\g Ind.Tip
\ge Indicative immediate past tense
\type s
\a e / [C] _ +/ _ [Ind.S1.S2]
\a we / [V] _ +/ _ [Ind.S1.S2]
\a a / [C] _ +/ _ [Ind.S3]
\a wa / [V] _ +/ _ [Ind.S3]
\a 0 +/ _ [Ind.S1.S2] +/ _ Ind.S3s
\u a
\o 80
\c SMV/SMV VR/SMV VR/SFV SFV/SFV

\g Ind.Tp
\ge Indicative present tense
\type s
\a 0 +/ _ [Ind.S1.S2.S3]
\a e / [C] _ +/ _ [Ind.S1.S2]
\a we / [V] _ +/ _ [Ind.S1.S2]
\a a / [C] _ +/ _ [Ind.S3]
\a wa / [V] _ +/ _ [Ind.S3]
\u 0
\o 80
\c SMV/SMV VR/SMV VR/SFV SFV/SFV

\g Ind.Trp
\ge Indicative remote past tense
\type s
\a e / [C] _ +/ _ [Ind.S1.S2]
\a we / [V] _ +/ _ [Ind.S1.S2]
\a a / [C] _ +/ _ [Ind.S3]
\a wa / [V] _ +/ _ [Ind.S3]
\u e
\o 80
\c SMV/SMV VR/SMV VR/SFV SFV/SFV

\g Inf
\ge Infinitive
\type s
\a e / [C] _
\a we / [V] _
\u e
\o 80
```

```
\c VR/Inf

\g Inf.Sd
\ge Infinitive subject
\type s
\a ru
\u ru
\o 90
\c Inf/Inf

\g Inf.Sp
\ge Infinitive plural subject
\type s
\a gu
\u gu
\o 90
\c Inf/Inf

\g Inf.Ss
\ge Infinitive singular subject
\type s
\a i
\u i
\o 90
\c Inf/Inf

\g inside
\ge inside
\type r
\a huwanya
\u huwanya
\c PNR

\g judge
\ge judge
\type r
\a haawe
\u haawe
\mp MC2
\c VR

\g just/only
\ge just/only
\type s
\a nga
\u nga
\c NW/NW RflPro/RflPro PN/PN Adj/Adj PosPro/PosPro
                          PerPro/PerPro Prep/Prep Deictic/Deictic
VR/VR

\g knife
\ge knife
\type r
\a kesebu
\u kesebu
\c NW

```

```
(1319)      \g large
(1320)      \ge large
(1321)      \type r
(1322)      \a anyakaro
(1323)      \u anyakaro
(1324)      \c Adj
(1325)
(1326)      \g leg/foot
(1327)      \ge leg/foot
(1328)      \type r
(1329)      \a afe
(1330)      \u afe
(1331)      \c PNR
(1332)
(1333)      \g leg/foot.3sP
(1334)      \ge his/her/its leg/foot
(1335)      \type r
(1336)      \a afo
(1337)      \u afo
(1338)      \c PNR
(1339)
(1340)      \g lie
(1341)      \ge lie
(1342)      \type r
(1343)      \a yaawa
(1344)      \u yaawa
(1345)      \mp MC2
(1346)      \c VR
(1347)
(1348)      \g light
(1349)      \ge light
(1350)      \type r
(1351)      \a lala
(1352)      \u lala_1
(1353)      \c NW
(1354)
(1355)      \g like.that
(1356)      \ge like.that
(1357)      \type r
(1358)      \a onou
(1359)      \u onou
(1360)      \c Adj
(1361)
(1362)      \g like1
(1363)      \ge like1
(1364)      \type s
(1365)      \a fe
(1366)      \u fe
(1367)      \o 100
(1368)      \c NW/NW Deictic/Deictic Adj/Adj
(1369)
(1370)      \g like2
(1371)      \ge like2
(1372)      \type s
(1373)      \a neu +/ this _
(1374)      \u neu
(1375)      \o 100
```

```
\c Deictic/Deictic

\g look.for
\ge look.for
\type r
\a kuru
\u kuru
\mp MC2
\c VR

\g man
\ge man
\type r
\a fai
\u fai
\c NW

\g many
\ge many
\type r
\a baingaro
\u baingaro
\c Adj

\g my
\ge my
\type r
\a yame
\u yame
\c PosPro

\g myself
\ge myself
\type r
\a yage
\u yage
\c RflPro

\g N.not
\ge not
\type r
\a uuwa
\u uuwa_1
\c NomNeg

\g Nom
\ge Nominalizer
\type s
\a ou / [C] _
\a wou / [V] _
\u ou
\o 110
\c VR/Nom Nom/Nom

\g nose
\ge nose
\type r
```

```
\a mutu
\u mutu
\c PNR

\g O1d
\ge to us (dual)
\type s
\a hiri
\a hir / _ [V]
\u hiri
\mp MC1
\o 30
\c VR/VR

\g O1p
\ge to us (plural)
\type s
\a higi
\a hig / _ [V]
\u higi
\mp MC1
\o 30
\c VR/VR

\g O1s
\ge to me
\type s
\a hi
\a hiy / _ [V]
\u hii
\mp MC2
\o 30
\c VR/VR

\g O2d
\ge to you (dual)
\type s
\a tiri
\a tir / _ [V]
\u tiri
\mp MC1
\o 30
\c VR/VR

\g O2p
\ge to you (plural)
\type s
\a tigi
\a tig / _ [V]
\u tigi
\mp MC1
\o 30
\c VR/VR

\g O2s
\ge to you (singular)
\type s
```

```
\a hini
\a hin / _ [V]
\u hini
\mp MC1
\o 30
\c VR/VR

\g O3d
\ge to them (dual)
\type s
\a niri
\a nir / _ [V]
\u niri
\mp MC1
\o 30
\c VR/VR

\g O3p
\ge to them (plural)
\type s
\a nigi
\a nig / _ [V]
\u nigi
\mp MC1
\o 30
\c VR/VR

\g O3s
\ge to him/her/it
\type s
\a tu
\u tu
\mp MC2
\o 30
\c VR/VR

\g ocean
\ge ocean
\type r
\a hiiri
\u hiiri
\c NW

\g one/a
\ge one/a
\type r
\a ete
\u ete
\c Adj

\g our.dual
\ge our.dual
\type r
\a heire
\u heire
\c PosPro

```

```
\g our.plural
\ge our.plural
\type r
\a ebere
\u ebere
\c PosPro

\g ourselves.dual
\ge ourselves.dual
\type r
\a hare
\u hare
\c RflPro

\g ourselves.plural
\ge ourselves.plural
\type r
\a hage
\u hage
\c RflPro

\g p1
\ge plural
\type s
\a gana
\u gana
\o 20
\c VR/VR

\g P1d
\ge first dual possessive
\type s
\a re
\u re
\o 10
\c PNR/PN

\g P1p
\ge first plural possessive
\type s
\a ge
\u ge
\o 10
\c PNR/PN

\g P1s
\ge first singular possessive
\type s
\a fe
\u fe
\o 10
\c PNR/PN

\g p2
\ge plural
\type s
\a ru
```

```
\u ru
\o 20
\c VR/VR

\g P2d
\ge second dual possessive
\type s
\a tere
\u tere
\o 10
\c PNR/PN

\g P2p
\ge second plural possessive
\type s
\a tege
\u tege
\o 10
\c PNR/PN

\g P2s
\ge second singular possessive
\type s
\a ne
\a te
\u ne
\o 10
\c PNR/PN

\g P3d
\ge third dual possessive
\type s
\a nere
\u nere
\o 10
\c PNR/PN

\g P3p
\ge third plural possessive
\type s
\a nege
\u nege
\o 10
\c PNR/PN

\g P3s
\ge third singular possessive
\type s
\a 0
\a u
\u 0
\o 10
\c PNR/PN

\g perf
\ge perfective
\type s
```

```
\a ha
\u ha
\o 100
\c SFV/SFVW

\g pertaining.to
\ge pertaining.to
\type s
\a ha
\u ha
\c Adj/Adj NW/NW Deictic/Deictic PosPro/PosPro

\g pig
\ge pig
\type r
\a buuwa
\u buuwa
\c NW

\g place
\ge place
\type r
\a ha
\a haumu
\u ha
\c NW

\g plant
\ge plant
\type r
\a tau
\a ta
\u tau
\mp MC2 RC1
\c VR

\g poke
\ge poke
\type r
\a tu / _ [O]
\a tum / _ [B]
\a tun / _ [A]
\a tung / _ [X]
\u tuN
\mp MC2
\c VR

\g pour.out
\ge pour.out
\type r
\a feu
\u feu
\mp MC2
\c VR

\g Prediction
\ge Prediction
```

```
\type s
\a ga
\u ga
\o 100
\c VSjS/VSjS

\g prepare
\ge prepare
\type r
\a haiya
\u haiya
\mp MC2
\c VR

\g press
\ge press
\type r
\a ilei
\u ilei
\mp MC2
\c VR

\g put
\ge put
\type r
\a tafa
\a taf / _ [V]
\u tafa
\mp MC1
\c VR

\g quickly
\ge quickly
\type r
\a fasa
\u fasa
\mp MC2
\c VR

\g quickly
\ge quickly
\type s
\a ya
\u ya
\c VR/VR

\g red
\ge red
\type r
\a giiri
\u giiri
\c Adj

\g remove
\ge remove
\type r
\a uulu
```

```
\u uulu
\mp MC2
\c VR

\g run.away
\ge run.away
\type r
\a faga
\u faga
\mp MC2
\c VR

\g see.O1d
\ge see us (dual)
\type r
\a herire
\a herir / _ [V]
\u herire
\mp MC1
\c VR

\g see.O1p
\ge see us (plural)
\type r
\a herige
\a herig / _ [V]
\u herige
\mp MC1
\c VR

\g see.O1s
\ge see me
\type r
\a yeriye
\a yeriy / _ [V]
\u yeriye
\mp MC1
\c VR

\g see.O2d
\ge see you (dual)
\type r
\a terire
\a terir / _ [V]
\u terire
\mp MC1
\c VR

\g see.O2p
\ge see you (plural)
\type r
\a terige
\a terig / _ [V]
\u terige
\mp MC1
\c VR

```

```
\g see.O2s
\ge see you (singular)
\type r
\a neriye
\a neriy / _ [V]
\u neriye
\mp MC1
\c VR

\g see.O3d
\ge see them (dual)
\type r
\a nerire
\a nerir / _ [V]
\u nerire
\mp MC1
\c VR

\g see.O3p
\ge see them (plural)
\type r
\a nerige
\a nerig / _ [V]
\u nerige
\mp MC1
\c VR

\g see.O3s
\ge see him/her/it
\type r
\a ure
\a ur / _ [V]
\u ure
\mp MC1
\c VR

\g send
\ge send
\type r
\a huuru
\u huuru
\mp MC2
\c VR

\g shine
\ge shine
\type r
\a lala
\u lala_2
\mp MC2
\c VR

\g short
\ge short
\type r
\a tutu
\u tutu
```

```
\c Adj

\g shove
\ge shove
\type r
\a nuu
\u nuu
\mp MC2
\c VR

\g show
\ge show
\type r
\a abiti
\u abiti
\mp MC2
\c VR

\g sibling.same.sex.his/her/its
\ge sibling.same.sex.his/her/its
\type r
\a bemu
\u bai_2
\c NW

\g sibling.same.sex.possessive
\ge sibling.same.sex.possessive
\type r
\a bai +/ ~_ P3s
\u bai_2
\c PNR

\g sibling.same.sex.referential
\ge sibling.same.sex.referential
\type r
\a babo
\u bai_2
\c NW

\g sit
\ge sit
\type r
\a yafa
\a yaf / _ [V]
\u yafa
\mp MC1
\c VR

\g Sj.S1/3
\ge subjunctive first or third subject
\type s
\a o / [C] _
\a wo / [V] _
\u o
\o 80
\c VR/VSj

```

```
\g Sj.S2
\ge subjunctive second subject
\type s
\a a / [C] _
\a wa / [V] _
\u a
\o 80
\c VR/VSj

\g Sj.Sd
\ge subjunctive dual subject
\type s
\a ru
\u ru
\o 90
\c VSj/VSjS

\g Sj.Sp
\ge subjunctive plural subject
\type s
\a gu
\u gu
\o 90
\c VSj/VSjS

\g Sj.Ss
\ge subjunctive singular subject
\type s
\a u
\u u
\o 90
\c VSj/VSjS

\g sleep.pl
\ge sleep.pl
\type r
\a jiri
\u jiri
\mp MC2
\c VR

\g small
\ge small
\type r
\a menekele
\a melekene
\u menekele
\c Adj

\g smooth
\ge smooth
\type r
\a fii
\u fii
\mp MC2
\c VR

```

```
\g some
\ge some
\type r
\a ila
\u ila
\c Adj

\g speak
\ge speak
\type r
\a le
\u le
\mp MC2
\c VR

\g SR
\ge switch referent
\type s
\a nga
\u nga
\o 110
\c SMV/SMVW Contra/SMVW

\g SR.TO
\ge switch referent with temporal overlap
\type s
\a g +/ _ [Ind.T] +/ _ [Contra.Subj]
\u g
\o 70
\c VR/SMV

\g SR.TS
\ge switch referent with temporal succession
\type s
\a 0 +/ _ [Ind.T] +/ _ [Contra.Subj]
\u 0
\o 70
\c VR/SMV

\g stand
\ge stand
\type r
\a oto
\u oto
\mp MC2
\c VR

\g stick.to
\ge stick.to
\type r
\a fada
\u fada
\mp MC2
\c VR

\g stone
\ge stone
```

```
\type r
\a megebu
\u megebu
\c NW

\g straight
\ge straight
\type r
\a tigini
\u tigini
\c Adj

\g talk/word
\ge talk/word
\type r
\a beele
\u beele
\c NW

\g teach
\ge teach
\type r
\a agotete
\u agotete
\mp MC2
\c VR

\g tear
\ge tear
\type r
\a korai
\u korai
\mp MC2
\c VR

\g tell
\ge tell
\type r
\a hai
\u hai
\mp MC2
\c VR

\g that
\ge that
\type r
\a wa
\u wa
\c Deictic

\g their.dual
\ge their.dual
\type r
\a neire
\u neire
\c PosPro

```

```
\g their.plural
\ge their.plural
\type r
\a nebere
\u nebere
\c PosPro

\g theirselves.dual
\ge theirselves.dual
\type r
\a nare
\u nare
\c RflPro

\g theirselves.plural
\ge theirselves.plural
\type r
\a nage
\u nage
\c RflPro

\g they
\ge they
\type r
\a nere
\u nere
\c PerPro

\g thigh
\ge thigh
\type r
\a gaigu
\u gaigu
\c NW

\g thing
\ge thing
\type r
\a dada
\u dada
\c NW

\g this
\ge this
\type r
\a be
\u be
\c Deictic

\g thought
\ge thought
\type r
\a nina
\u nina
\c PNR

\g throw
```

```
\ge throw
\type r
\a aya
\u aya
\mp MC2
\c VR

\g TO
\ge temporal overlap
\type s
\a 0
\u 0
\o 70
\c VR/SMVW

\g true
\ge true
\type r
\a ngalenga
\u ngalenga
\c Adj

\g TS
\ge temporal succession
\type s
\a du
\u du
\o 70
\c VR/SMVW

\g turn.back.on
\ge turn.back.on
\type r
\a moge
\u moge
\mp MC2
\c VR

\g untie
\ge untie
\type r
\a fuutu
\u fuutu
\mp MC2
\c VR

\g V.not
\ge not
\type r
\a ada
\u ada
\c VNeg

\g vAux.S1d
\ge verbal auxiliary first dual subject
\type s
\a yare
```

```
\u yare
\o 90
\c vAuxR/vAuxW

\g vAux.S1p
\ge verbal auxiliary first plural subject
\type s
\a ya
\u ya
\o 90
\c vAuxR/vAuxW

\g vAux.S1s
\ge verbal auxiliary first singular subject
\type s
\a ni
\u ni
\o 90
\c vAuxR/vAuxW

\g vAux.S3d
\ge verbal auxiliary third dual subject
\type s
\a dere
\u dere
\o 90
\c vAuxR/vAuxW

\g vAux.S3p
\ge verbal auxiliary third plural subject
\type s
\a de
\u de
\o 90
\c vAuxR/vAuxW

\g vAux.S3s
\ge verbal auxiliary third singular subject
\type s
\a na
\u na
\o 90
\c vAuxR/vAuxW

\g vAux.Tf.S2d
\ge verbal auxiliary second dual subject
\type s
\a nadere
\u nadere
\o 90
\c vAuxR/vAuxW

\g vAux.Tf.S2p
\ge verbal auxiliary second plural subject
\type s
\a nade
\u nade
```

```
\o 90
\c vAuxR/vAuxW

\g vAux.Tf.S2s
\ge verbal auxiliary second singular subject, future tense
\type s
\a nahe
\u nahe
\o 90
\c vAuxR/vAuxW

\g vAux.Tnf.S2d
\ge verbal auxiliary second dual subject, non-future tense
\type s
\a yare
\u yare
\o 90
\c vAuxR/vAuxW

\g vAux.Tnf.S2p
\ge verbal auxiliary second plural subject, non-future tense
\type s
\a ya
\u ya
\o 90
\c vAuxR/vAuxW

\g vAux.Tnf.S2s
\ge verbal auxiliary second singular subject, non-future
\type s
\a ne
\u ne
\o 90
\c vAuxR/vAuxW

\g vAuxR
\ge verbal auxiliary root
\type r
\a bo
\u bo
\c vAuxR

\g very
\ge very
\type r
\a bai
\u bai
\c Adj

\g village
\ge village
\type r
\a matane
\u matane
\c NW

\g walk
```

```
\ge walk
\type r
\a golo
\u golo
\mp MC2
\c VR

\g watch
\ge watch
\type r
\a fele
\u fele
\mp MC2
\c VR

\g watch.over
\ge watch.over
\type r
\a oofa
\u oofa
\mp MC2
\c VR

\g we
\ge we
\type r
\a ere
\u ere
\c PerPro

\g well
\ge well
\type s
\a de
\u de
\c VR/VR

\g what
\ge what
\type r
\a taate
\u taate
\c IP

\g wild
\ge wild
\type r
\a kui
\u kui
\c Adj

\g with
\ge with
\type r
\a bagu +/ ~_ causative
\u bagu
\c Prep
```

```

\g with.dual
\ge with.dual
\type r
\a ngare
\u ngare
\c Prep

\g without
\ge without
\type r
\a uwau
\u uwau
\c Prep

\g woman
\ge woman
\type r
\a aita
\u aita
\c NW

\g work
\ge work
\type r
\a haruwe
\u haruwe
\c NW

\g write
\ge write
\type r
\a yerege
\a yereg / _ [V]
\u yerege
\mp MC1
\c VR

\g yell.to/shout
\ge yell.to/shout
\type r
\a yaure
\u yaure
\mp MC2
\c VR

\g you.plural
\ge you.plural
\type r
\a tere
\u tere
\c PerPro

\g you.sing
\ge you.sing
\type r
\a ne
```

```
\u ne
\c PerPro

\g your.dual
\ge your.dual
\type r
\a teire
\u teire
\c PosPro

\g your.plural
\ge your.plural
\type r
\a tebere
\u tebere
\c PosPro

\g your.sing
\ge your.sing
\type r
\a name
\u name
\c PosPro

\g yourselves.dual
\ge yourselves.dual
\type r
\a tare
\u tare
\c RflPro

\g yourselves.plural
\ge yourselves.plural
\type r
\a tage
\u tage
\c RflPro
```

APPENDIX 3

THE OGEA DATR LEXICON

A3.1 Overview

This appendix provides listings of the following: The Ogea DATR "theory" file (the lexicon), the Ogea DATR query file, a DOS batch file that invokes DATR, and the result of running DATR using the query file--a query trace file.

The Ogea DATR theory file is the heart of the Ogea DATR lexicon. It is divided into the following sections: function library, interfaces, AMPLE environment constraints, templates, categories, English, Tok Pisin, and Ogea. The Ogea DATR query file is the driver for the interface between DATR and AMPLE. The entries in the query file mirror the entries in the Ogea AMPLE database file. The DOS batch file listing contains two calls to DATR. The first call creates a token file, the second executes the query file against the token file. The Ogea DATR trace file is the output of running the DOS batch file. The trace file provides the Ogea AMPLE lexical entries, combining both roots and suffixes.

A3.2 The Ogea DATR Theory File

A3.2.1 Description

The Ogea DATR theory file is divided into the following sections: function library, interfaces, AMPLE environment constraints, templates, categories, English, Tok Pisin, and Ogea.

The function library loads the standard DATR function library and adds a new function called 'chop'. The 'chop' function is used in the application of various Ogea morphophonemic rules.

The interfaces section provides a one-way interface from DATR to AMPLE and a dictionary. The AMPLE interface supports generation of AMPLE lexical files that use either English, Ogea, or Tok Pisin as the value for the record marker. The dictionary interface generates dictionary entries using plain text labels for each field rather than codes.

The AMPLE environment constraints are a series of nodes that act as variables. Each node defines the detail of a specific string or morpheme constraint. The string constraints begin with a node labeled 'SEC', and the morpheme constriants with a node labeled 'MEC'. These two abstract nodes define the AMPLE makers used to indicate the start of a constraint.

The templates section provides a series of abstract nodes used for two purposes. First, they provide support for multiple allomorph entries in the case that no morphophonemic rules are applied. Second, they provide a mechanism by which morphophonemic rules can be applied to underlying forms to produce the correct allomorphic form. Through the application of morphophonemic rules, only a single underlying form needs to be specified. This reduces lexicon maintenance, and also allows testing of morphophonemic rules.

The categories section provides abstract nodes that represent grammatical and morphological categories. It is through these abstract nodes that much of the redundancy of the Ogea AMPLE lexicon can be reduced.

The English section provides a set of nodes ordered by English glosses. This section supports DATR queries that utilize English words to retrieve the appropriate Ogea. As such it is the entry point into the lexicon through English.

The Tok Pisin section is the entry point into the lexicon through Tok Pisin. Whereas the English section contains a full set of English nodes, the Tok Pisin section has only three nodes and was developed merely to illustrate how queries may start with Tok Pisin and retrieve Ogea.

The last section is the most comprehensive and is the Ogea section. This section contains the full set of nodes for Ogea roots and suffixes. All entries in the Ogea section are uniquely identified by Ogea morphemes and allomorphs. The morphemes and allomorphs are identified as roots or suffixes by a prefix (R_ or S_)[20]. In the case of homophones, a suffix is added to uniquely identify that morpheme or allomorph. The suffix sometimes uses a numeric identifier, and sometimes uses a grammatical gloss. Note that in the case of homophones, an abstract entry is provided to support a query that is ignorant of the uniquely identifying suffixes. For example, the root node `R_anyakaro` may be queried without knowledge of the two entries for `anyakaro`. A query will be redirected to `R_anyakaro_1` and `R_anayakaro_2`.

A3.2.2 File Listing

Filename: ogea.dtr

Note: line numbers added for reference only. Not in actual file.

[20] DATR node names may not start with a numeral, so zero morphs may not be entered with as a node name starting with '0'. For this reason and pure management of the file, a decision was made to use a leading prefix for Ogea node names.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                            %
% File:            Ogea.dtr                                  %
% Purpose:         Lexicon for Ogea Language                 %
% Ogea:            A Papuan Language of Papua New Guinea     %
% Author:          Michael Colburn                           %
% Email:           mcolburn@sprintmail.com                   %
% Created:         October 9, 1999                           %
% Last update:     November 13, 1999                         %
% Version:         1.0                                       %
%                                                            %
% Notes:    Supports queries from/to English, Ogea,          %
%           and Tok Pisin                                    %
%                                                            %
%                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%  Function Library %%%%%%%%%%%%%%%%%%%%%%%%
%

#load 'std.dtr' all.

% Chop: Chops off the last byte of a string
Chop:
<> ==
<$a> == Implode:<Reverse:<Rest:<Reverse:<Explode:<$a>>>>>.

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%  Interfaces  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

I_AMPLE:
% Generates records with English as first field
<amp lex eng> ==     <amp gloss eng>
                     <amp gloss morpheme name>
%                    <amp gloss national>
                     <amp rest>
% Generates records with the national language as first field
<amp lex nat> ==     <amp gloss national>
                     <amp gloss morpheme name>
                     <amp gloss eng>
                     <amp rest>
% Generates records with the vernacular language
% as first field
<amp lex ver> ==     <amp gloss vernacular>
                     <amp gloss morpheme name>
                     <amp gloss eng>
                     <amp gloss national>
                     <amp rest>
% Generates records with the morphname as the first field
<amp lex morname> == <amp gloss morpheme name>
                     <amp gloss eng>
%                    <amp gloss national>
```

```
                     <amp rest>
<amp rest> ==        <amp type>
                     "<lex form lex>"
                     <amp underlying form>
                     <amp allo property>
                     <amp order class>
                     <amp category>
                     <amp elsewhere allomorph>
                     <amp mcc>
                     <amp features>
                     <amp infix location>
                     <amp noload>
                     <amp comment>'\n'
<amp allomorph> == \a' ' "<allo form>"' ' "<allo prop>"' '
                               "<allo sec>"' ' "<allo
mec>"'\n'
<amp category> == \c' ' "<lex cat>"'\n'
<amp comment> == \co' ' "<lex com>"'\n'
<amp elsewhere allomorph> == \e' ' "<lex else allo>"'\n'
<amp features> == \f' ' "<lex feat>"'\n'
<amp gloss eng> == \ge' ' "<lex gloss eng>" '\n'  % not called
                                                    since used
as mor name
<amp gloss vernacular> == \gv' ' "<lex under>" '\n'
<amp gloss national> == \gn' ' "<lex gloss nat>" '\n'
<amp infix location> == \loc' ' "<lex loc>" '\n'
<amp gloss morpheme name> == \g' ' "<lex gloss eng morname>"
'\n'
<amp allo property> == \mp' ' "<lex prop>"'\n'
<amp noload> == \_no' ' "<lex noload>"'\n'
<amp order class> == \o' ' "<lex order class>"'\n'
<amp mcc> == \mcc' ' "<lex mcc>"'\n'
<amp type> == \type' ' "<lex type>"'\n'
<amp underlying form> == \u' ' "<lex under>"'\n'.

I_DICT:
<dict lex eng> ==  "<lex gloss eng>"'\n'
                    '  Ogea:         ' "<lex under>"'\n'
                    '  Category:     '"<lex cat>"'\n'
                    '  Property:     ' "<lex prop>"'\n'
                    '  Allomorph(s): \n'
                    "<lex form dict>"'\n'
<dict allomorph> ==
             '                '"<allo form>"'\n'
 '                  Property:     ' "<allo prop>"'\n'
 '                  String Environment:    ' "<allo sec>"'\n'
 '                  Morpheme Environment:    ' "<allo
mec>"'\n'.

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%% Templates %%%%%%%%%%%%%%%%%%%%%%%%%%%
%

ALLO1:
<lex form lex> == <lex form lex allo1>
```

```
<lex form lex allo1> == C_DLT
<lex form dict> == <lex form dict allo1>
<lex form dict allo1> == C_DLT.

ALLO2:
<lex form lex> == <lex form lex allo2>
<lex form lex allo2> == C_DLT
<lex form dict> == <lex form dict allo2>
<lex form dict allo2> == C_DLT.

ALLO3:
<lex form lex> == <lex form lex allo3>
<lex form lex allo3> == C_DLT
<lex form dict> == <lex form dict allo3>
<lex form dict allo3> == C_DLT.

ALLO4:
<lex form lex> == <lex form lex allo4>
<lex form lex allo4> == C_DLT
<lex form dict> == <lex form dict allo4>
<lex form dict allo4> == C_DLT.

ALLO5:
<lex form lex> == <lex form lex allo5>
<lex form lex allo5> == C_DLT
<lex form dict> == <lex form dict allo5>
<lex form dict allo5> == C_DLT.

ALLOMORPH:
<allo form default> == "<phon under>"
<allo form elision> == Chop:<"<phon under>">
<allo form epenthetic_w> == Implode:<w "<phon under>">
<allo form glide> == Implode:<Chop:<"<phon under>"> y>
<allo form redup> ==  Implode:<First:<Explode:
                                  <"<phon under>">>
First:<Rest:<Explode:
                                  <"<phon under>">>>>
<allo form syncope> == Chop:<"<phon under>">
<allo form n_b> == Implode:<Chop:<"<phon under>"> m>
<allo form n_a> == Implode:<Chop:<"<phon under>"> n>
<allo form n_x> == Implode:<Chop:<"<phon under>"> ng>
<allo form n_v> == Chop:<"<phon under>">
<allo prop> ==
<allo mec> ==
<allo sec> ==
<allo sec epenthetic_w> == "SEC_02"
<allo sec glide> == "SEC_01"
<allo sec syncope> == "SEC_01"
<allo sec n_b> == "SEC_04"
<allo sec n_a> == "SEC_05"
<allo sec n_x> == "SEC_07"
<allo sec n_v> == "SEC_06".

DEFAULT:
<lex form lex> == <lex form lex default>
<lex form dict> == <lex form dict default>
<lex form lex default> == C_DLT
```

```
<lex form dict default> == C_DLT.

ELISION:
<lex form lex> == <lex form lex elision>
<lex form lex elision> == C_DLT
<lex form dict> == <lex form dict elision>
<lex form dict elision> == C_DLT.

EPENTHETIC_W:
<lex form lex> == <lex form lex epenthetic_w>
<lex form lex epenthetic_w> == C_DLT
<lex form dict> == <lex form dict epenthetic_w>
<lex form dict epenthetic_w> == C_DLT.

GLIDE:
<lex form lex> == <lex form lex glide>
<lex form lex glide> == C_DLT
<lex form dict> == <lex form dict glide>
<lex form dict glide> == C_DLT.

NASAL:
<lex form lex> == <lex form lex n_v> <lex form lex n_b>
                              <lex form lex n_a> <lex form lex n_x>
<lex form lex n_b> == C_DLT
<lex form lex n_a> == C_DLT
<lex form lex n_x> == C_DLT
<lex form lex n_v> == C_DLT
<lex form dict> == <lex form dict n_v> <lex form dict n_b>
                               <lex form dict n_a> <lex form dict n_x>
                               <lex form dict n_b> == C_DLT
<lex form dict n_a> == C_DLT
<lex form dict n_x> == C_DLT
<lex form dict n_v> == C_DLT.

REDUP:
<lex form lex> == <lex form lex redup>
<lex form lex redup> == C_DLT
<lex form dict> == <lex form dict redup>
<lex form dict redup> == C_DLT.

SYNCOPE:
<lex form dict> == <lex form dict syncope>
<lex form dict syncope> == C_DLT
<lex form lex> == <lex form lex syncope>
<lex form lex syncope> == C_DLT.

% Default Lexical Template
C_DLT:
<lex form lex> == I_AMPLE:<amp allomorph>
<lex form dict > == I_DICT:<dict allomorph>.

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


```

```
%%%%%%%%%%%%%%%%%%%%%%%% AMPLE Environment Constraints %%%%%%
%
% String Environment Constraints

SEC:
<sec environ> == Implode:<'/ ' "<environ>"' '>
<allo sec> == <sec environ.>.   % if don't cut path,
                                            must specify each
extension

SEC_01:
<> == SEC
<environ> == '_ [V]'.

SEC_02:
<> == SEC
<environ> == '[V] _'.

SEC_03:
<> == SEC
<environ> == '[C] _'.

SEC_04:
<> == SEC
<environ> == '_ [B]'.

SEC_05:
<> == SEC
<environ> == '_ [A]'.

SEC_06:
<> == SEC
<environ> == '_ [O]'.

SEC_07:
<> == SEC
<environ> == '_ [X]'.

SEC_08:
<> == SEC
<environ> == 'o _ #'.

SEC_09:
<> == SEC
<environ> == 'e _'.

SEC_10:
<> == SEC
<environ> ==      'o _'.

SEC_11:
<> == SEC
<environ> == 'a _'.

% Morpheme Environment Constraints

MEC:
```

```
<mec environ> == Implode:<'+/ ' "<environ>"' '>
<allo mec> == <mec environ.>.   % if don't cut path,
                                                must specify each
extension

MEC_01:
<> == MEC
<environ> == '~_ TO'.

MEC_02:
<> == MEC
<environ> == '~_ P3s'.

MEC_03:
<> == MEC
<environ> == '~_ causative'.

MEC_04:
<> == MEC
<environ> == '_ [Contra.Subj]'.

MEC_05:
<> == MEC
<environ> == '_ Contra'.

MEC_06:
<> == MEC
<environ> == 'Ind.Tf _ ~SR'.

MEC_07:
<> == MEC
<environ> == 'Ind.Tf _ SR'.

MEC_08:
<> == MEC
<environ> == '[Ind.Tip.Tp] _ SR'.

MEC_09:
<> == MEC
<environ> == 'Ind.Trp _'.

MEC_10:
<> == MEC
<environ> == '[Ind.Tip.Tp] _'.

MEC_11:
<> == MEC
<environ> == 'Ind.Tip _'.

MEC_12:
<> == MEC
<environ> == 'Ind.Tp _ ~SR'.

MEC_13:
<> == MEC
<environ> == 'Ind.Tp _ SR'.

```

```
MEC_14:
<> == MEC
<environ> == 'Ind.Tf _'.

MEC_15:
<> == MEC
<environ> == 'Ind.Tip _ ~SR'.

MEC_16:
<> == MEC
<environ> == '_ [Ind.S1.S2]'.

MEC_17:
<> == MEC
<environ> == '_ Ind.S3s'.

MEC_18:
<> == MEC
<environ> == '_ [Ind.S1]'.

MEC_19:
<> == MEC
<environ> == '_ [Ind.S2]'.

MEC_20:
<> == MEC
<environ> == '_ [Ind.S1.S3]'.

MEC_21:
<> == MEC
<environ> == '_ [Ind.S3]'.

MEC_22:
<> == MEC
<environ> == '_ [Ind.S1.S2.S3]'.

MEC_23:
<> == MEC
<environ> == '_ [Ind.T]'.

MEC_24:
<> == MEC
<environ> == 'this _'.

MEC_25:
<> == MEC
<environ> == '[Contra.Subj] _'.



%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%% Categories %%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This section contains abstract entries that are
% language dependent
```

```

C_MORPHEME:
<> ==
<amp lex> == I_AMPLE
<dict lex> == I_DICT
<allo> == ALLOMORPH
<lex under> == "<phon under>"
<lex prop> ==
<lex noload> ==
<lex order class> ==
<lex form> == DEFAULT
<lex cat> ==
<lex loc> ==
<lex gloss nat> == 'No national language entry found'
<lex gloss eng> == 'No English entry found'
<lex gloss eng morname> == 'No English morph name found'.

C_INFIX:
<> == C_MORPHEME
<lex type> == i.

C_PREFIX:
<> == C_MORPHEME
<lex type> == p.

C_ROOT:
<> == C_MORPHEME
<lex type> == r.

C_R_Adj:
<> == C_ROOT
<lex cat> == 'Adj'.

C_R_Adj_VR:
<> == C_R_VR
<lex under> == Implode: <"<phon under>" _2>
<allo mec> == "MEC_01".

C_R_Adv:
<> == C_ROOT
<lex cat> == 'Adv'.

C_R_Deictic:
<> == C_ROOT
<lex cat> == 'Deictic'.

C_R_IP:
<> == C_ROOT
<lex cat> == 'IP'.

C_R_NomNeg:
<> == C_ROOT
<lex cat> == 'NomNeg'.

C_R_NW:
<> == C_ROOT
<lex cat> == 'NW'.
```

```

C_R_PerPro:
<> == C_ROOT
<lex cat> == 'PerPro'.

C_R_PNR:
<> == C_ROOT
<lex cat> == 'PNR'.

C_R_PosPro:
<> == C_ROOT
<lex cat> == 'PosPro'.

C_R_Prep:
<> == C_ROOT
<lex cat> == 'Prep'.

C_R_RflPro:
<> == C_ROOT
<lex cat> == 'RflPro'.

C_R_vAuxR:
<> == C_ROOT
<lex cat> == 'vAuxR'.

C_R_VNeg:
<> == C_ROOT
<lex cat> == 'VNeg'.

C_R_VR:
<> == C_ROOT
<lex prop> == 'MC2'
<lex cat> == 'VR'.

C_R_VR_MC1:
<> == C_R_VR
<lex form> == DEFAULT SYNCOPE
<lex prop> == 'MC1'.

C_R_VR_R:
<> == C_R_VR
<lex form> == DEFAULT REDUP
<lex prop> == 'MC2 RC1'.

C_SUFFIX:
<> == C_MORPHEME
<lex type> == s.

C_S_ASPECT:
<> == C_SUFFIX.

C_S_ATTRIB:
<> == C_S_EPENTHETIC
<lex order class> == 110
<lex cat> == 'Adj/Atr'.

C_S_BENEFACTIVE:
```

```
<> == C_SUFFIX
<lex prop> == 'MC2'
<lex order class> == 60
<lex cat> == 'VR/VR'.

C_S_BENEFACTIVE_MC1:
<> == C_S_BENEFACTIVE
<lex form> == DEFAULT SYNCOPE
<lex prop> == 'MC1'.

C_S_CONTRA:
<> == C_SUFFIX.

C_S_CONTRA_SUBJ:
<> == C_S_CONTRA
<lex form> == DEFAULT EPENTHETIC_W
<allo sec default> == "SEC_03"
<allo mec> == "MEC_05"
<lex order class> == 80
<lex cat> == 'VR/Contra SMV/Contra'.

C_S_DIFF_REF_TEMPORALITY:
<> == C_S_TEMPORALITY
<allo mec> == "MEC_23" "MEC_04"
<lex cat> == 'VR/SMV'.

C_S_EPENTHETIC:
<> == C_SUFFIX
<lex form> == DEFAULT EPENTHETIC_W
<allo sec default> == "SEC_03".

C_S_IND_SUBJ:
<> == C_SUFFIX
<lex order class> == 90
<lex cat> == 'SFV/SFV SFV/SFVW SMV/SMV'.

C_S_IND_TENSE:
<> == C_SUFFIX
<allo sec default> == "SEC_03"
<lex form> == DEFAULT EPENTHETIC_W
<lex order class> == 80
<lex cat> == 'SMV/SMV VR/SMV VR/SFV SFV/SFV'.

C_S_INF:
<> == C_SUFFIX.

C_S_INF_SUBJ:
<> == C_S_INF
<lex order class> == 90
<lex cat> == 'Inf/Inf'.

C_S_OBJECT:
<> == C_SUFFIX
<lex prop> == 'MC2'
<lex order class> == 30
<lex cat> == 'VR/VR'.

```

```
C_S_OBJECT_MC1:
<> == C_S_OBJECT
<lex form> == DEFAULT SYNCOPE
<lex prop> == 'MC1'.

C_S_PLURAL:
<> == C_SUFFIX
<lex order class> == 20
<lex cat> == 'VR/VR'.

C_S_PRONOUN_POSS:
<> == C_SUFFIX
<lex order class> == 10
<lex cat> == 'PNR/PN'.

C_S_SAME_REF_TEMPORALITY:
<> == C_S_TEMPORALITY
<lex cat> == 'VR/SMVW'.

C_S_SUBJUNCTIVE:
<> == C_SUFFIX.

C_S_SUBJUNCTIVE_SUBJ_NUMBER:
<> == C_S_SUBJUNCTIVE
<lex order class> == 90
<lex cat> == 'VSj/VSjS'.

C_S_SUBJUNCTIVE_SUBJ_PERSON:
<> == C_S_SUBJUNCTIVE
<lex form> == DEFAULT EPENTHETIC_W
<lex order class> == 80
<lex cat> == 'VR/VSj'
<allo sec default> == "SEC_03".

C_S_TEMPORALITY:
<> == C_S_VERBAL_SUFFIX
<lex order class> == 70.

C_S_V_AUX:
<> == C_SUFFIX
<lex order class> == 90
<lex cat> == 'vAuxR/vAuxW'.

C_S_VERBAL_SUFFIX:
<> == C_SUFFIX
<lex cat> == 'VR/VR'.

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%% English %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

ENG_able:
<> == "R_edo"
<lex gloss eng> == 'able'.
```

```

ENG_allow/leave:
<> == "R_awa"
<lex gloss eng> == 'allow/leave'.

ENG_always:
<> == "R_gaigai"
<lex gloss eng> == 'always'.

ENG_animal:
<> == "R_kaafa"
<lex gloss eng> == 'animal'.

ENG_appear/arrive:
<> == "R_heige"
<lex gloss eng> == 'appear/arrive'.

ENG_ask:
<> == "R_isoki"
<lex gloss eng> == 'ask'.

ENG_at:
<> == "R_la"
<lex gloss eng> == 'at'.

ENG_Attrib:
<> == "S_ou_Attrib"
<lex gloss eng morname> == 'Attrib'
<lex gloss eng> == 'Attributive marker'.

ENG_B1d:
<> == "S_hari"
<lex gloss eng morname> == 'B1d'
<lex gloss eng> == 'for us (dual)'.

ENG_B1p:
<> == "S_hagi"
<lex gloss eng morname> == 'B1p'
<lex gloss eng> == 'for us (plural)'.

ENG_B1s:
<> == "S_hai"
<lex gloss eng morname> == 'B1s'
<lex gloss eng> == 'for me'.

ENG_B2d:
<> == "S_tari"
<lex gloss eng morname> == 'B2d'
<lex gloss eng> == 'for you (dual)'.

ENG_B2p:
<> == "S_tagi"
<lex gloss eng morname> == 'B2p'
<lex gloss eng> == 'for you (plural)'.

ENG_B2s:
<> == "S_hani"
```

```
<lex gloss eng morname> == 'B2s'
<lex gloss eng> == 'for you (singular)'.

ENG_B3d:
<> == "S_nari"
<lex gloss eng morname> == 'B3d'
<lex gloss eng> == 'for them (dual)'.

ENG_B3p:
<> == "S_nagi"
<lex gloss eng morname> == 'B3p'
<lex gloss eng> == 'for them (plural)'.

ENG_B3s:
<> == "S_fu"
<lex gloss eng morname> == 'B3s'
<lex gloss eng> == 'for him/her/it'.

ENG_bad:
<> == "R_inyaba_1"
<lex gloss eng> == 'bad'.

ENG_be:
<> == "R_inyi"
<lex gloss eng> == 'be'.

ENG_become_bad:
<> == "R_inyaba_2"
<lex gloss eng> == 'become.bad'.

ENG_become_crooked:
<> == "R_kobibi_2"
<lex gloss eng> == 'become.crooked'.

ENG_become_good:
<> == "R_hilou_2"
<lex gloss eng> == 'become.good'.

ENG_become_heavy:
<> == "R_ou_2"
<lex gloss eng> == 'become.heavy'.

ENG_become_large:
<> == "R_anyakaro_2"
<lex gloss eng> == 'become.large'.

ENG_become_many:
<> == "R_baingaro_2"
<lex gloss eng> == 'become.many'.

ENG_become_nothing:
<> == "R_uuwa_2"
<lex gloss eng> == 'become.nothing'.

ENG_become_red:
<> == "R_giiri_2"
<lex gloss eng> == 'become.red'.
```

```

ENG_become_short:
<> == "R_tutu_2"
<lex gloss eng> == 'become.short'.

ENG_become_small:
<> == "R_menekele_2"
<lex gloss eng> == 'become.small'.

ENG_become_straight:
<> == "R_tigini_2"
<lex gloss eng> == 'become.straight'.

ENG_become_true:
<> == "R_ngalenga_2"
<lex gloss eng> == 'become.true'.

ENG_become_wild:
<> == "R_kui_2"
<lex gloss eng> == 'become.wild'.

ENG_bound:
<> == "R_dage"
<lex gloss eng> == 'bound'.

ENG_break_in_two:
<> == "R_kou"
<lex gloss eng> == 'break.in.two'.

ENG_causative:
<> == "S_mu"
<lex gloss eng> == 'causative'.

ENG_certainty:
<> == "S_nga_2"
<lex gloss eng> == 'certainty'.

ENG_chase:
<> == "R_soo"
<lex gloss eng> == 'chase'.

ENG_cloth:
<> == "R_ogola"
<lex gloss eng> == 'cloth'.

ENG_come_down:
<> == "R_me"
<lex gloss eng> == 'come.down'.

ENG_come_up:
<> == "R_mai"
<lex gloss eng> == 'come.up'.

ENG_completely:
<> == "S_boro"
<lex gloss eng> == 'completely'.

```

```
ENG_connect/join:
<> == "R_bagumu"
<lex gloss eng> == 'connect/join'.

ENG_consume:
<> == "R_nyi"
<lex gloss eng> == 'consume'.

ENG_Contra:
<> == "S_ge_1"
<lex gloss eng morname> == 'Contra'
<lex gloss eng> == 'Contrafactual'.

ENG_Contra_S1d:
<> == "S_edere"
<lex gloss eng morname> == 'Contra.S1d'
<lex gloss eng> == 'Contractual first dual subject'.

ENG_Contra_S1p:
<> == "S_ede"
<lex gloss eng morname> == 'Contra.S1p'
<lex gloss eng> == 'Contractual first plural subject'.

ENG_Contra_S1s:
<> == "S_e_Contra"
<lex gloss eng morname> == 'Contra.S1s'
<lex gloss eng> == 'Contractual first singular subject'.

ENG_Contra_S2/3s/d/p:
<> == "S_a_1"
<lex gloss eng morname> == 'Contra.S2/3s/d/p'
<lex gloss eng> == 'Contractual second or third singular,
                                dual, or plural subject'.

ENG_crooked:
<> == "R_kobibi_1"
<lex gloss eng> == 'crooked'.

ENG_cut:
<> == "R_karu"
<lex gloss eng> == 'cut'.

ENG_deflect:
<> == "R_ariri"
<lex gloss eng> == 'deflect'.

ENG_desire:
<> == "R_gau"
<lex gloss eng> == 'desire'.

ENG_die:
<> == "R_ume"
<lex gloss eng> == 'die'.

ENG_different:
<> == "R_fere"
<lex gloss eng> == 'different'.
```

```

ENG_do_not:
<> == "R_adai"
<lex gloss eng> == 'do.not'.

ENG_do_thus:
<> == "R_ono"
<lex gloss eng> == 'do.thus'.

ENG_do_work:
<> == "R_haruwe_2"
<lex gloss eng> == 'do.work'.

ENG_dream:
<> == "R_wewe"
<lex gloss eng> == 'dream'.

ENG_encircle:
<> == "R_kati"
<lex gloss eng> == 'encircle'.

ENG_face:
<> == "R_hogo"
<lex gloss eng> == 'face'.

ENG_fear:
<> == "R_umuge"
<lex gloss eng> == 'fear'.

ENG_fight:
<> == "R_wara"
<lex gloss eng> == 'fight'.

ENG_fill:
<> == "R_bolo"
<lex gloss eng> == 'fill'.

ENG_finish:
<> == "R_usu"
<lex gloss eng> == 'finish'.

ENG_fly:
<> == "R_fililai"
<lex gloss eng> == 'fly'.

ENG_follow:
<> == "R_oojo"
<lex gloss eng> == 'follow'.

ENG_gather:
<> == "R_elege"
<lex gloss eng> == 'gather'.

ENG_get/take:
<> == "R_te"
<lex gloss eng> == 'get/take'.

```

```
ENG_get_O1d:
<> == "R_haire"
<lex gloss eng morname> == 'get.O1d'
<lex gloss eng> == 'get us (dual)'.

ENG_get_O1p:
<> == "R_haige"
<lex gloss eng morname> == 'get.O1p'
<lex gloss eng> == 'get us (plural)'.

ENG_get_O1s:
<> == "R_haife"
<lex gloss eng morname> == 'get.O1s'
<lex gloss eng> == 'get me'.

ENG_get_O2d:
<> == "R_hatire"
<lex gloss eng morname> == 'get.O2d'
<lex gloss eng> == 'get you (dual)'.

ENG_get_O2p:
<> == "R_hatige"
<lex gloss eng morname> == 'get.O2p'
<lex gloss eng> == 'get you (plural)'.

ENG_get_O2s:
<> == "R_haine"
<lex gloss eng morname> == 'get.O2s'
<lex gloss eng> == 'get you (singular)'.

ENG_get_O3d:
<> == "R_hanire"
<lex gloss eng morname> == 'get.O3d'
<lex gloss eng> == 'get them (dual)'.

ENG_get_O3p:
<> == "R_hanige"
<lex gloss eng morname> == 'get.O3p'
<lex gloss eng> == 'get them (plural)'.

ENG_get_O3s:
<> == "R_hau"
<lex gloss eng morname> == 'get.O3s'
<lex gloss eng> == 'get him/her/it'.

ENG_get_up:
<> == "R_jai"
<lex gloss eng> == 'get.up'.

ENG_give_O1d:
<> == "R_hiri"
<lex gloss eng morname> == 'give.O1d'
<lex gloss eng> == 'give us (dual)'.

ENG_give_O1p:
<> == "R_higi"
<lex gloss eng morname> == 'give.O1p'
```

```
<lex gloss eng> == 'give us (plural)'.

ENG_give_O1s:
<> == "R_hii"
<lex gloss eng morname> == 'give.O1s'
<lex gloss eng> == 'give me'.

ENG_give_O2d:
<> == "R_tiri"
<lex gloss eng morname> == 'give.O2d'
<lex gloss eng> == 'give you (dual)'.

ENG_give_O2p:
<> == "R_tigi"
<lex gloss eng morname> == 'give.O2p'
<lex gloss eng> == 'give you (plural)'.

ENG_give_O2s:
<> == "R_hini"
<lex gloss eng morname> == 'give.O2s'
<lex gloss eng> == 'give you (singular)'.

ENG_give_O3d:
<> == "R_niri"
<lex gloss eng morname> == 'give.O3d'
<lex gloss eng> == 'give them (dual)'.

ENG_give_O3p:
<> == "R_nigi"
<lex gloss eng morname> == 'give.O3p'
<lex gloss eng> == 'give them (plural)'.

ENG_give_O3s:
<> == "R_tu"
<lex gloss eng morname> == 'give.O3s'
<lex gloss eng> == 'give him/her/it'.

ENG_go:
<> == "R_goi"
<lex gloss eng> == 'go'.

ENG_go_down:
<> == "R_mini"
<lex gloss eng> == 'go.down'.

ENG_good:
<> == "R_hilou"
<lex gloss eng> == 'good'.

ENG_Hab:
<> == "S_g_1"
<lex gloss eng morname> == 'Hab'
<lex gloss eng> == 'Habitual'.

ENG_hand:
<> == "R_owo"
<lex gloss eng> == 'hand'.
```

```

ENG_he/she/it:
<> == "R_no"
<lex gloss eng> == 'he/she/it'.

ENG_hear/understand/obey:
<> == "R_isi"
<lex gloss eng> == 'hear/understand/obey'.

ENG_heavy:
<> == "R_ou_1"
<lex gloss eng> == 'heavy'.

ENG_help:
<> == "R_hoyo"
<lex gloss eng> == 'help'.

ENG_himself/herself/itself:
<> == "R_nogo"
<lex gloss eng> == 'himself/herself/itself'.

ENG_his/her/its:
<> == "R_nomo"
<lex gloss eng> == 'his/her/its'.

ENG_hit_O1d:
<> == "R_harire"
<lex gloss eng morname> == 'hit.O1d'
<lex gloss eng> == 'hit us (dual)'.

ENG_hit_O1p:
<> == "R_harige"
<lex gloss eng morname> == 'hit.O1p'
<lex gloss eng> == 'hit us (plural)'.

ENG_hit_O1s:
<> == "R_yari"
<lex gloss eng morname> == 'hit.O1s'
<lex gloss eng> == 'hit me'.

ENG_hit_O2d:
<> == "R_tarire"
<lex gloss eng morname> == 'hit.O2d'
<lex gloss eng> == 'hit you (dual)'.

ENG_hit_O2p:
<> == "R_tarige"
<lex gloss eng morname> == 'hit.O2p'
<lex gloss eng> == 'hit you (plural)'.

ENG_hit_O2s:
<> == "R_nari"
<lex gloss eng morname> == 'hit.O2s'
<lex gloss eng> == 'hit you (singular)'.

ENG_hit_O3d:
<> == "R_narire"
```

```
<lex gloss eng morname> == 'hit.O3d'
<lex gloss eng> == 'hit them (dual)'.

ENG_hit_O3p:
<> == "R_narige"
<lex gloss eng morname> == 'hit.O3p'
<lex gloss eng> == 'hit them (plural)'.

ENG_hit_O3s:
<> == "R_wari"
<lex gloss eng morname> == 'hit.O3s'
<lex gloss eng> == 'hit him/her/it'.

ENG_I:
<> == "R_ji"
<lex gloss eng> == 'I'.

ENG_idle:
<> == "R_ewe"
<lex gloss eng> == 'idle'.

ENG_Ind_S1d:
<> == "S_Ind_S1d"
<lex gloss eng morname> == 'Ind.S1d'
<lex gloss eng> == 'Indicative first dual subject'.

ENG_Ind_S1p:
<> == "S_Ind_S1p"
<lex gloss eng morname> == 'Ind.S1p'
<lex gloss eng> == 'Indicative first plural subject'.

ENG_Ind_S1s:
<> == "S_Ind_S1s"
<lex gloss eng morname> == 'Ind.S1s'
<lex gloss eng> == 'Indicative first singular subject'.

ENG_Ind_S2d:
<> == "S_Ind_S2d"
<lex gloss eng morname> == 'Ind.S2d'
<lex gloss eng> == 'Indicative second dual subject'.

ENG_Ind_S2p:
<> == "S_Ind_S2p"
<lex gloss eng morname> == 'Ind.S2p'
<lex gloss eng> == 'Indicative second plural subject'.

ENG_Ind_S2s:
<> == "S_Ind_S2s"
<lex gloss eng morname> == 'Ind.S2s'
<lex gloss eng> == 'Indicative second singular subject'.

ENG_Ind_S3d:
<> == "S_Ind_S3d"
<lex gloss eng morname> == 'Ind.S3d'
<lex gloss eng> == 'Indicative third dual subject'.

ENG_Ind_S3p:
```

```
<> == "S_Ind_S3p"
<lex gloss eng morname> == 'Ind.S3p'
<lex gloss eng> == 'Indicative third plural subject'.

ENG_Ind_S3s:
<> == "S_Ind_S3s"
<lex gloss eng morname> == 'Ind.S3s'
<lex gloss eng> == 'Indicative third singular subject'.

ENG_Ind_Tf:
<> == "S_Ind_Tf"
<lex gloss eng morname> == 'Ind.Tf'
<lex gloss eng> == 'Indicative future tense'.

ENG_Ind_Tip:
<> == "S_Ind_Tip"
<lex gloss eng morname> == 'Ind.Tip'
<lex gloss eng> == 'Indicative immediate past tense'.

ENG_Ind_Tp:
<> == "S_Ind_Tp"
<lex gloss eng morname> == 'Ind.Tp'
<lex gloss eng> == 'Indicative present tense'.

ENG_Ind_Trp:
<> == "S_Ind_Trp"
<lex gloss eng morname> == 'Ind.Trp'
<lex gloss eng> == 'Indicative remote past tense'.

ENG_Inf:
<> == "S_e_Inf"
<lex gloss eng morname> == 'Inf'
<lex gloss eng> == 'Infinitive'.

ENG_Inf_Sd:
<> == "S_ru_Inf"
<lex gloss eng morname> == 'Inf.Sd'
<lex gloss eng> == 'Infinitive subject'.

ENG_Inf_Sp:
<> == "S_gu_Inf"
<lex gloss eng morname> == 'Inf.Sp'
<lex gloss eng> == 'Infinitive plural subject'.

ENG_Inf_Ss:
<> == "S_i_Inf"
<lex gloss eng morname> == 'Inf.Ss'
<lex gloss eng> == 'Infinitive singular subject'.

ENG_inside:
<> == "R_huwanya"
<lex gloss eng> == 'inside'.

ENG_judge:
<> == "R_haawe"
<lex gloss eng> == 'judge'.
```

```
ENG_just/only:
<> == "S_nga_1"
<lex gloss eng> == 'just/only'.

ENG_knife:
<> == "R_kesebu"
<lex gloss eng> == 'knife'.

ENG_large:
<> == "R_anyakaro_1"
<lex gloss eng> == 'large'.

ENG_leg/foot:
<> == "R_afe"
<lex gloss eng> == 'leg/foot'.

ENG_leg/foot_3sP:
<> == "R_afo"
<lex gloss eng morname> == 'leg/foot.3sP'
<lex gloss eng> == 'his/her/its leg/foot'.

ENG_lie:
<> == "R_yaawa"
<lex gloss eng> == 'lie'.

ENG_light:
<> == "R_lala_1"
<lex gloss eng> == 'light'.

ENG_like_that:
<> == "R_onou"
<lex gloss eng> == 'like.that'.

ENG_like1:
<> == "S_fe_1"
<lex gloss eng> == 'like1'.

ENG_like2:
<> == "S_neu"
<lex gloss eng> == 'like2'.

ENG_look_for:
<> == "R_kuru"
<lex gloss eng> == 'look.for'.

ENG_man:
<> == "R_fai"
<lex gloss eng> == 'man'.

ENG_many:
<> == "R_baingaro_1"
<lex gloss eng> == 'many'.

ENG_my:
<> == "R_yame"
<lex gloss eng> == 'my'.

```

```
ENG_myself:
<> == "R_yage"
<lex gloss eng> == 'myself'.

ENG_N_not:
<> == "R_uuwa_1"
<lex gloss eng morname> == 'N.not'
<lex gloss eng> == 'not'.

ENG_Nom:
<> == "S_ou_Nom"
<lex gloss eng morname> == 'Nom'
<lex gloss eng> == 'Nominalizer'.

ENG_nose:
<> == "R_mutu"
<lex gloss eng> == 'nose'.

ENG_O1d:
<> == "S_hiri"
<lex gloss eng morname> == 'O1d'
<lex gloss eng> == 'to us (dual)'.

ENG_O1p:
<> == "S_higi"
<lex gloss eng morname> == 'O1p'
<lex gloss eng> == 'to us (plural)'.

ENG_O1s:
<> == "S_hii"
<lex gloss eng morname> == 'O1s'
<lex gloss eng> == 'to me'.

ENG_O2d:
<> == "S_tiri"
<lex gloss eng morname> == 'O2d'
<lex gloss eng> == 'to you (dual)'.

ENG_O2p:
<> == "S_tigi"
<lex gloss eng morname> == 'O2p'
<lex gloss eng> == 'to you (plural)'.

ENG_O2s:
<> == "S_hini"
<lex gloss eng morname> == 'O2s'
<lex gloss eng> == 'to you (singular)'.

ENG_O3d:
<> == "S_niri"
<lex gloss eng morname> == 'O3d'
<lex gloss eng> == 'to them (dual)'.

ENG_O3p:
<> == "S_nigi"
<lex gloss eng morname> == 'O3p'
<lex gloss eng> == 'to them (plural)'.
```

```

ENG_O3s:
<> == "S_tu"
<lex gloss eng morname> == 'O3s'
<lex gloss eng> == 'to him/her/it'.

ENG_ocean:
<> == "R_hiiri"
<lex gloss eng> == 'ocean'.

ENG_one/a:
<> == "R_ete"
<lex gloss eng> == 'one/a'.

ENG_our_dual:
<> == "R_heire"
<lex gloss eng> == 'our.dual'.

ENG_our_plural:
<> == "R_ebere"
<lex gloss eng> == 'our.plural'.

ENG_ourselves_dual:
<> == "R_hare"
<lex gloss eng> == 'ourselves.dual'.

ENG_ourselves_plural:
<> == "R_hage"
<lex gloss eng> == 'ourselves.plural'.

ENG_p1:
<> == "S_gana"
<lex gloss eng morname> == 'p1'
<lex gloss eng> == 'plural'.

ENG_P1d:
<> == "S_re_2"
<lex gloss eng morname> == 'P1d'
<lex gloss eng> == 'first dual possessive'.

ENG_P1p:
<> == "S_ge_2"
<lex gloss eng morname> == 'P1p'
<lex gloss eng> == 'first plural possessive'.

ENG_P1s:
<> == "S_fe_2"
<lex gloss eng morname> == 'P1s'
<lex gloss eng> == 'first singular possessive'.

ENG_p2:
<> == "S_ru_1"
<lex gloss eng morname> == 'p2'
<lex gloss eng> == 'plural'.

ENG_P2d:
<> == "S_tere"
```

```
<lex gloss eng morname> == 'P2d'
<lex gloss eng> == 'second dual possessive'.

ENG_P2p:
<> == "S_tege"
<lex gloss eng morname> == 'P2p'
<lex gloss eng> == 'second plural possessive'.

ENG_P2s:
<> == "S_ne_2"
<lex gloss eng morname> == 'P2s'
<lex gloss eng> == 'second singular possessive'.

ENG_P3d:
<> == "S_nere"
<lex gloss eng morname> == 'P3d'
<lex gloss eng> == 'third dual possessive'.

ENG_P3p:
<> == "S_nege"
<lex gloss eng morname> == 'P3p'
<lex gloss eng> == 'third plural possessive'.

ENG_P3s:
<> == "S_0_Pro_Poss"
<lex gloss eng morname> == 'P3s'
<lex gloss eng> == 'third singular possessive'.

ENG_perf:
<> == "S_ha_1"
<lex gloss eng morname> == 'perf'
<lex gloss eng> == 'perfective'.

ENG_pertaining_to:
<> == "S_ha_2"
<lex gloss eng> == 'pertaining.to'.

ENG_pig:
<> == "R_buuwa"
<lex gloss eng> == 'pig'.

ENG_place:
<> == "R_ha"
<lex gloss eng> == 'place'.

ENG_plant:
<> == "R_tau"
<lex gloss eng> == 'plant'.

ENG_poke:
<> == "R_tuN"
<lex gloss eng> == 'poke'.

ENG_pour_out:
<> == "R_feu"
<lex gloss eng> == 'pour.out'.

```

```
ENG_Prediction:
<> == "S_ga"
<lex gloss eng> == 'Prediction'.

ENG_prepare:
<> == "R_haiya"
<lex gloss eng> == 'prepare'.

ENG_press:
<> == "R_ilei"
<lex gloss eng> == 'press'.

ENG_put:
<> == "R_tafa"
<lex gloss eng> == 'put'.

ENG_quickly:
<> == "R_fasa"
<lex gloss eng> == 'quickly'.

ENG_quickly_2:
<> == "S_ya_Manner"
<lex gloss eng> == 'quickly'.

ENG_red:
<> == "R_giiri_1"
<lex gloss eng> == 'red'.

ENG_remove:
<> == "R_uulu"
<lex gloss eng> == 'remove'.

ENG_run_away:
<> == "R_faga"
<lex gloss eng> == 'run.away'.

ENG_see_O1d:
<> == "R_herire"
<lex gloss eng morname> == 'see.O1d'
<lex gloss eng> == 'see us (dual)'.

ENG_see_O1p:
<> == "R_herige"
<lex gloss eng morname> == 'see.O1p'
<lex gloss eng> == 'see us (plural)'.

ENG_see_O1s:
<> == "R_yeriye"
<lex gloss eng morname> == 'see.O1s'
<lex gloss eng> == 'see me'.

ENG_see_O2d:
<> == "R_terire"
<lex gloss eng morname> == 'see.O2d'
<lex gloss eng> == 'see you (dual)'.

ENG_see_O2p:
```

```
<> == "R_terige"
<lex gloss eng morname> == 'see.O2p'
<lex gloss eng> == 'see you (plural)'.

ENG_see_O2s:
<> == "R_neriye"
<lex gloss eng morname> == 'see.O2s'
<lex gloss eng> == 'see you (singular)'.

ENG_see_O3d:
<> == "R_nerire"
<lex gloss eng morname> == 'see.O3d'
<lex gloss eng> == 'see them (dual)'.

ENG_see_O3p:
<> == "R_nerige"
<lex gloss eng morname> == 'see.O3p'
<lex gloss eng> == 'see them (plural)'.

ENG_see_O3s:
<> == "R_ure"
<lex gloss eng morname> == 'see.O3s'
<lex gloss eng> == 'see him/her/it'.

ENG_send:
<> == "R_huuru"
<lex gloss eng> == 'send'.

ENG_shine:
<> == "R_lala_2"
<lex gloss eng> == 'shine'.

ENG_short:
<> == "R_tutu_1"
<lex gloss eng> == 'short'.

ENG_shove:
<> == "R_nuu"
<lex gloss eng> == 'shove'.

ENG_show:
<> == "R_abiti"
<lex gloss eng> == 'show'.

ENG_sibling_same_sex_his/her/its:
<> == "R_bemu"
<lex gloss eng> == 'sibling.same.sex.his/her/its'.

ENG_sibling_same_sex_possessive:
<> == "R_bai_2"
<lex gloss eng> == 'sibling.same.sex.possessive'.

ENG_sibling_same_sex_referential:
<> == "R_babo"
<lex gloss eng> == 'sibling.same.sex.referential'.

ENG_sit:
```

```
<> == "R_yafa"
<lex gloss eng> == 'sit'.

ENG_Sj_S1/3:
<> == "S_o_Subj"
<lex gloss eng morname> == 'Sj.S1/3'
<lex gloss eng> == 'subjunctive first or third subject'.

ENG_Sj_S2:
<> == "S_a_2"
<lex gloss eng morname> == 'Sj.S2'
<lex gloss eng> == 'subjunctive second subject'.

ENG_Sj_Sd:
<> == "S_ru_2"
<lex gloss eng morname> == 'Sj.Sd'
<lex gloss eng> == 'subjunctive dual subject'.

ENG_Sj_Sp:
<> == "S_gu_Subj"
<lex gloss eng morname> == 'Sj.Sp'
<lex gloss eng> == 'subjunctive plural subject'.

ENG_Sj_Ss:
<> == "S_u"
<lex gloss eng morname> == 'Sj.Ss'
<lex gloss eng> == 'subjunctive singular subject'.

ENG_sleep_pl:
<> == "R_jiri"
<lex gloss eng> == 'sleep.pl'.

ENG_small:
<> == "R_menekele_1"
<lex gloss eng> == 'small'.

ENG_smooth:
<> == "R_fii"
<lex gloss eng> == 'smooth'.

ENG_some:
<> == "R_ila"
<lex gloss eng> == 'some'.

ENG_speak:
<> == "R_le"
<lex gloss eng> == 'speak'.

ENG_SR:
<> == "S_nga_3"
<lex gloss eng morname> == 'SR'
<lex gloss eng> == 'switch referent'.

ENG_SR_TO:
<> == "S_g_2"
<lex gloss eng morname> == 'SR.TO'
<lex gloss eng> == 'switch referent with temporal overlap'.
```

```

ENG_SR_TS:
<> == "S_0_1"
<lex gloss eng morname> == 'SR.TS'
<lex gloss eng> == 'switch referent with temporal succession'.

ENG_stand:
<> == "R_oto"
<lex gloss eng> == 'stand'.

ENG_stick_to:
<> == "R_fada"
<lex gloss eng> == 'stick.to'.

ENG_stone:
<> == "R_megebu"
<lex gloss eng> == 'stone'.

ENG_straight:
<> == "R_tigini"
<lex gloss eng> == 'straight'.

ENG_talk/word:
<> == "R_beele"
<lex gloss eng> == 'talk/word'.

ENG_teach:
<> == "R_agotete"
<lex gloss eng> == 'teach'.

ENG_tear:
<> == "R_korai"
<lex gloss eng> == 'tear'.

ENG_tell:
<> == "R_hai"
<lex gloss eng> == 'tell'.

ENG_that:
<> == "R_wa"
<lex gloss eng> == 'that'.

ENG_their_dual:
<> == "R_neire"
<lex gloss eng> == 'their.dual'.

ENG_their_plural:
<> == "R_nebere"
<lex gloss eng> == 'their.plural'.

ENG_theirselves_dual:
<> == "R_nare"
<lex gloss eng> == 'theirselves.dual'.

ENG_theirselves_plural:
<> == "R_nage"
<lex gloss eng> == 'theirselves.plural'.
```

```

ENG_they:
<> == "R_nere"
<lex gloss eng> == 'they'.

ENG_thigh:
<> == "R_gaigu"
<lex gloss eng> == 'thigh'.

ENG_thing:
<> == "R_dada"
<lex gloss eng> == 'thing'.

ENG_this:
<> == "R_be"
<lex gloss eng> == 'this'.

ENG_thought:
<> == "R_nina"
<lex gloss eng> == 'thought'.

ENG_throw:
<> == "R_aya"
<lex gloss eng> == 'throw'.

ENG_TO:
<> == "S_0_2"
<lex gloss eng morname> == 'TO'
<lex gloss eng> == 'temporal overlap'.

ENG_true:
<> == "R_ngalenga_1"
<lex gloss eng> == 'true'.

ENG_TS:
<> == "S_du"
<lex gloss eng morname> == 'TS'
<lex gloss eng> == 'temporal succession'.

ENG_turn_back_on:
<> == "R_moge"
<lex gloss eng> == 'turn.back.on'.

ENG_untie:
<> == "R_fuutu"
<lex gloss eng> == 'untie'.

ENG_V_not:
<> == "R_ada"
<lex gloss eng morname> == 'V.not'
<lex gloss eng> == 'not'.

ENG_vAux_S1d:
<> == "S_yare_1_vAux"
<lex gloss eng morname> == 'vAux.S1d'
<lex gloss eng> == 'verbal auxiliary first dual subject'.

```

```
ENG_vAux_S1p:
<> == "S_ya_1_vAux"
<lex gloss eng morname> == 'vAux.S1p'
<lex gloss eng> == 'verbal auxiliary first plural subject'.

ENG_vAux_S1s:
<> == "S_ni_vAux"
<lex gloss eng morname> == 'vAux.S1s'
<lex gloss eng> == 'verbal auxiliary first singular subject'.

ENG_vAux_S3d:
<> == "S_dere_vAux"
<lex gloss eng morname> == 'vAux.S3d'
<lex gloss eng> == 'verbal auxiliary third dual subject'.

ENG_vAux_S3p:
<> == "S_de_vAux"
<lex gloss eng morname> == 'vAux.S3p'
<lex gloss eng> == 'verbal auxiliary third plural subject'.

ENG_vAux_S3s:
<> == "S_na_2"
<lex gloss eng morname> == 'vAux.S3s'
<lex gloss eng> == 'verbal auxiliary third singular subject'.

ENG_vAux_Tf_S2d:
<> == "S_nadere"
<lex gloss eng morname> == 'vAux.Tf.S2d'
<lex gloss eng> == 'verbal auxiliary second dual subject'.

ENG_vAux_Tf_S2p:
<> == "S_nade"
<lex gloss eng morname> == 'vAux.Tf.S2p'
<lex gloss eng> == 'verbal auxiliary second plural subject'.

ENG_vAux_Tf_S2s:
<> == "S_nahe"
<lex gloss eng morname> == 'vAux.Tf.S2s'
<lex gloss eng> == 'verbal auxiliary second singular subject,
                                future tense'.

ENG_vAux_Tnf_S2d:
<> == "S_yare_2_vAux"
<lex gloss eng morname> == 'vAux.Tnf.S2d'
<lex gloss eng> == 'verbal auxiliary second dual subject,
                                non-future tense'.

ENG_vAux_Tnf_S2p:
<> == "S_ya_2_vAux"
<lex gloss eng morname> == 'vAux.Tnf.S2p'
<lex gloss eng> == 'verbal auxiliary second plural subject,
                                non-future tense'.

ENG_vAux_Tnf_S2s:
<> == "S_ne_vAux"
<lex gloss eng morname> == 'vAux.Tnf.S2s'
<lex gloss eng> == 'verbal auxiliary second singular subject,
                                non-future'.

ENG_vAuxR:
<> == "R_bo"
<lex gloss eng morname> == 'vAuxR'
<lex gloss eng> == 'verbal auxiliary root'.

ENG_very:
<> == "R_bai_1"
<lex gloss eng> == 'very'.

ENG_village:
<> == "R_matane"
<lex gloss eng> == 'village'.

ENG_walk:
<> == "R_golo"
<lex gloss eng> == 'walk'.

ENG_watch:
<> == "R_fele"
<lex gloss eng> == 'watch'.

ENG_watch_over:
<> == "R_oofa"
<lex gloss eng> == 'watch.over'.

ENG_we:
<> == "R_ere"
<lex gloss eng> == 'we'.

ENG_well:
<> == "S_de_1"
<lex gloss eng> == 'well'.

ENG_what:
<> == "R_taate"
<lex gloss eng> == 'what'.

ENG_wild:
<> == "R_kui_1"
<lex gloss eng> == 'wild'.

ENG_with:
<> == "R_bagu"
<lex gloss eng> == 'with'.

ENG_with_dual:
<> == "R_ngare"
<lex gloss eng> == 'with.dual'.

ENG_without:
<> == "R_uwau"
<lex gloss eng> == 'without'.

ENG_woman:
<> == "R_aita"
```

```
<lex gloss eng> == 'woman'.

ENG_work:
<> == "R_haruwe"
<lex gloss eng> == 'work'.

ENG_write:
<> == "R_yerege"
<lex gloss eng> == 'write'.

ENG_yell_to/shout:
<> == "R_yaure"
<lex gloss eng> == 'yell.to/shout'.

ENG_you_plural:
<> == "R_tere"
<lex gloss eng> == 'you.plural'.

ENG_you_sing:
<> == "R_ne"
<lex gloss eng> == 'you.sing'.

ENG_your_dual:
<> == "R_teire"
<lex gloss eng> == 'your.dual'.

ENG_your_plural:
<> == "R_tebere"
<lex gloss eng> == 'your.plural'.

ENG_your_sing:
<> == "R_name"
<lex gloss eng> == 'your.sing'.

ENG_yourselves_dual:
<> == "R_tare"
<lex gloss eng> == 'yourselves.dual'.

ENG_yourselves_plural:
<> == "R_tage"
<lex gloss eng> == 'yourselves.plural'.


%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Tok Pisin %%%%%%%%%%%%%%%%%%%%%
%


TP_inap:
<> == "R_edo"
<lex gloss nat> == inap.

TP_lek:
<> == "R_afe"
<lex gloss nat> == lek.
```

```

TP_no:
<> == "R_ada"
<lex gloss nat> == no.

TP_no_ken:
<> == "R_adai"
<lex gloss nat> == 'no ken'.

TP_soim:
<> == "R_abiti"
<lex gloss nat> == soim.


%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Ogea %%%%%%%%%%%%%%%%%%%%%%%%%%
%

R_abiti:
<> == C_R_VR
<phon under> == abiti
<lex gloss eng> == ENG_show
<lex gloss nat> == TP_soim.

R_ada:
<> == C_R_VNeg
<phon under> == ada
<lex gloss eng> == ENG_V_not
<lex gloss nat> == TP_no.

R_adai:
<> == C_R_Adv
<phon under> == adai
<lex gloss eng> == ENG_do_not
<lex gloss nat> == TP_no_ken.

R_afe:
<> == C_R_PNR
<phon under> == afe
<lex gloss eng> == ENG_leg/foot
<lex gloss nat> == TP_lek.

R_afo:
<> == C_R_PNR
<phon under> == afo
<lex gloss eng> == ENG_leg/foot_3sP.

R_agotete:
<> == C_R_VR
<phon under> == agotete
<lex gloss eng> == ENG_teach.

R_aita:
<> == C_R_NW
<phon under> == aita
```

```
<lex gloss eng> == ENG_woman.

R_anyakaro:
<> == "R_anyakaro_1" "R_anyakaro_2".

R_anyakaro_1:
<> == C_R_Adj
<phon under> == anyakaro
<lex gloss eng> == ENG_large.


R_anyakaro_2:
<> == C_R_Adj_VR
<phon under> == anyakaro
<lex gloss eng> == ENG_become_large.

R_ariri:
<> == C_R_VR
<phon under> == ariri
<lex gloss eng> == ENG_deflect.

R_awa:
<> == C_R_VR_MC1
<phon under> == awa
<lex gloss eng> == ENG_allow/leave.

R_aya:
<> == C_R_VR
<phon under> == aya
<lex gloss eng> == ENG_throw.

R_babo:
<> == C_R_NW
<phon under> == babo
<lex gloss eng> == ENG_sibling_same_sex_referential
        <lex under> == bai_2.

R_bagu:
<> == C_R_Prep
<phon under> == bagu
<lex gloss eng> == ENG_with
        <allo mec> == "MEC_03".

R_bagumu:
<> == C_R_VR
<phon under> == bagumu
<lex gloss eng> == ENG_connect/join
        <lex under> == bagu-mu.

R_bai:
<> == "R_bai_1" "R_bai_2".

R_bai_1:
<> == C_R_Adj
<phon under> == bai
<lex gloss eng> == ENG_very.

```

```
R_bai_2:
<> == C_R_PNR
<phon under> == bai
<lex gloss eng> == ENG_sibling_same_sex_possessive
<lex under> ==    bai_2
<allo mec> == "MEC_02".

R_baingaro:
<> == "R_baingaro_1" "R_baingaro_2".

R_baingaro_1:
<> == C_R_Adj
<phon under> == baingaro
<lex gloss eng> == ENG_many.

R_baingaro_2:
<> == C_R_Adj_VR
<phon under> == baingaro
<lex gloss eng> == ENG_become_many.

R_be:
<> == C_R_Deictic
<phon under> == be
<lex gloss eng> == ENG_this.

R_beele:
<> == C_R_NW
<phon under> == beele
<lex gloss eng> == ENG_talk/word.

R_bemu:
<> == C_R_NW
<phon under> == bemu
<lex gloss eng> == ENG_sibling_same_sex_his/her/its
<lex under> == bai_2.

R_bo:
<> == C_R_vAuxR
<phon under> == bo
<lex gloss eng> == ENG_vAuxR.

R_bolo:
<> == C_R_VR
<phon under> == bolo
<lex gloss eng> == ENG_fill.

R_buuwa:
<> == C_R_NW
<phon under> == buuwa
<lex gloss eng> == ENG_pig.

R_dada:
<> == C_R_NW
<phon under> == dada
<lex gloss eng> == ENG_thing.

R_dage:
```

```
<> == C_R_VR_MC1
<phon under> == dage
<lex gloss eng> == ENG_bound.

R_ebere:
<> == C_R_PosPro
<phon under> == ebere
<lex gloss eng> == ENG_our_plural.

R_edo:
<> == C_R_Adv
<phon under> == edo
<lex gloss eng> == ENG_able
<lex gloss nat> == TP_inap.

R_elege:
<> == C_R_VR_MC1
<phon under> == elege
<lex gloss eng> == ENG_gather.

R_ere:
<> == C_R_PerPro
<phon under> == ere
<lex gloss eng> == ENG_we.

R_ete:
<> == C_R_Adj
<phon under> == ete
<lex gloss eng> == ENG_one/a.

R_ewe:
<> == C_R_Adj
<phon under> == ewe
<lex gloss eng> == ENG_idle.

R_fada:
<> == C_R_VR
<phon under> == fada
<lex gloss eng> == ENG_stick_to.

R_faga:
<> == C_R_VR
<phon under> == faga
<lex gloss eng> == ENG_run_away.

R_fai:
<> == C_R_NW
<phon under> == fai
<lex gloss eng> == ENG_man.

R_fasa:
<> == C_R_VR
<phon under> == fasa
<lex gloss eng> == ENG_quickly.

R_fele:
<> == C_R_VR
```

```
<phon under> == fele
<lex gloss eng> == ENG_watch.

R_fere:
<> == C_R_Adj
<phon under> == fere
<lex gloss eng> == ENG_different.

R_feu:
<> == C_R_VR
<phon under> == feu
<lex gloss eng> == ENG_pour_out.

R_fii:
<> == C_R_VR
<phon under> == fii
<lex gloss eng> == ENG_smooth.

R_fililai:
<> == C_R_VR
<phon under> == fililai
<lex gloss eng> == ENG_fly.

R_fuutu:
<> == C_R_VR
<phon under> == fuutu
<lex gloss eng> == ENG_untie.

R_gaigai:
<> == C_R_Adv
<phon under> == gaigai
<lex gloss eng> == ENG_always.

R_gaigu:
<> == C_R_NW
<phon under> == gaigu
<lex gloss eng> == ENG_thigh.

R_gau:
<> == C_R_NW
<phon under> == gau
<lex gloss eng> == ENG_desire.

R_giiri:
<> == "R_giiri_1" "R_giiri_2".

R_giiri_1:
<> == C_R_Adj
<phon under> == giiri
<lex gloss eng> == ENG_red.

R_giiri_2:
<> == C_R_Adj_VR
<phon under> == giiri
<lex gloss eng> == ENG_become_red.

R_goi:
```

```
<> == C_R_VR
<phon under> == goi
<lex gloss eng> == ENG_go
<lex form> == DEFAULT GLIDE.

R_golo:
<> == C_R_VR
<phon under> == golo
<lex gloss eng> == ENG_walk.

R_ha:
<> == C_R_NW
<phon under> == ha
<lex gloss eng> == ENG_place
<lex form> == DEFAULT ALLO2
        <allo form allo2> == haumu.

R_haawe:
<> == C_R_VR
<phon under> == haawe
<lex gloss eng> == ENG_judge.

R_hage:
<> == C_R_RflPro
<phon under> == hage
<lex gloss eng> == ENG_ourselves_plural.

R_hai:
<> == C_R_VR
<phon under> == hai
<lex gloss eng> == ENG_tell.

R_haife:
<> == C_R_VR_MC1
<phon under> == haife
<lex gloss eng> == ENG_get_O1s.

R_haige:
<> == C_R_VR_MC1
<phon under> == haige
<lex gloss eng> == ENG_get_O1p.

R_haine:
<> == C_R_VR_MC1
<phon under> == haine
<lex gloss eng> == ENG_get_O2s.

R_haire:
<> == C_R_VR_MC1
<phon under> == haire
<lex gloss eng> == ENG_get_O1d.

R_haiya:
<> == C_R_VR
<phon under> == haiya
<lex gloss eng> == ENG_prepare.
```

```
R_hanige:
<> == C_R_VR_MC1
<phon under> == hanige
<lex gloss eng> == ENG_get_O3p.

R_hanire:
<> == C_R_VR_MC1
<phon under> == hanire
<lex gloss eng> == ENG_get_O3d.

R_hare:
<> == C_R_RflPro
<phon under> == hare
<lex gloss eng> == ENG_ourselves_dual.

R_harige:
<> == C_R_VR_MC1
<phon under> == harige
<lex gloss eng> == ENG_hit_O1p.

R_harire:
<> == C_R_VR_MC1
<phon under> == harire
<lex gloss eng> == ENG_hit_O1d.

R_haruwe:
<> == C_R_NW
<phon under> == haruwe
<lex gloss eng> == ENG_work.

R_haruwe_2:
<> == C_R_VR
<phon under> == haruwe
<lex gloss eng> == ENG_do_work
<lex under> == haruwe_2
<allo mec> == "MEC_01".

R_hatige:
<> == C_R_VR_MC1
<phon under> == hatige
<lex gloss eng> == ENG_get_O2p.

R_hatire:
<> == C_R_VR_MC1
<phon under> == hatire
<lex gloss eng> == ENG_get_O2d.

R_hau:
<> == C_R_VR
<phon under> == hau
<lex gloss eng> == ENG_get_O3s.

R_heige:
<> == C_R_VR_MC1
<phon under> == heige
<lex gloss eng> == ENG_appear/arrive.
```

```
R_heire:
<> == C_R_PosPro
<phon under> == heire
<lex gloss eng> == ENG_our_dual.

R_herige:
<> == C_R_VR_MC1
<phon under> == herige
<lex gloss eng> == ENG_see_O1p.

R_herire:
<> == C_R_VR_MC1
<phon under> == herire
<lex gloss eng> == ENG_see_O1d.

R_higi:
<> == C_R_VR_MC1
<phon under> == higi
<lex gloss eng> == ENG_give_O1p.

R_hii:
<> == C_R_VR
<phon under> == hii
<lex gloss eng> == ENG_give_O1s
<lex form> == DEFAULT GLIDE.

R_hiiri:
<> == C_R_NW
<phon under> == hiiri
<lex gloss eng> == ENG_ocean.

R_hilou:
<> == C_R_Adj
<phon under> == hilou
<lex gloss eng> == ENG_good
<lex form> == DEFAULT ALLO2
<allo form allo2> == hiloo.

R_hilou_2:
<> == C_R_Adj_VR
<phon under> == hilou
<lex gloss eng> == ENG_become_good
<lex form> == DEFAULT ALLO2
<allo form allo2> == hiloo
<allo mec allo2> == "MEC_01".

R_hini:
<> == C_R_VR_MC1
<phon under> == hini
<lex gloss eng> == ENG_give_O2s.

R_hiri:
<> == C_R_VR_MC1
<phon under> == hiri
<lex gloss eng> == ENG_give_O1d.

R_hogo:
```

```
<> == C_R_PNR
<phon under> == hogo
<lex gloss eng> == ENG_face.

R_hoyo:
<> == C_R_VR
<phon under> == hoyo
<lex gloss eng> == ENG_help.

R_huuru:
<> == C_R_VR
<phon under> == huuru
<lex gloss eng> == ENG_send.

R_huwanya:
<> == C_R_PNR
<phon under> == huwanya
<lex gloss eng> == ENG_inside.

R_ila:
<> == C_R_Adj
<phon under> == ila
<lex gloss eng> == ENG_some.

R_ilei:
<> == C_R_VR
<phon under> == ilei
<lex gloss eng> == ENG_press.

R_inyaba:
<> == "R_inyaba_1" "R_inyaba_2".

R_inyaba_1:
<> == C_R_Adj
<phon under> == inyaba
<lex gloss eng> == ENG_bad.

R_inyaba_2:
<> == C_R_Adj_VR
<phon under> == inyaba
<lex gloss eng> == ENG_become_bad.

R_inyi:
<> == C_R_VR_MC1
<phon under> == inyi
<lex gloss eng> == ENG_be.

R_isi:
<> == C_R_VR_MC1
<phon under> == isi
<lex gloss eng> == ENG_hear/understand/obey.

R_isoki:
<> == C_R_VR
<phon under> == isoki
<lex gloss eng> == ENG_ask.

```

```
R_jai:
<> == C_R_VR
<phon under> == jai
<lex gloss eng> == ENG_get_up
<lex form> == DEFAULT GLIDE.

R_ji:
<> == C_R_PerPro
<phon under> == ji
<lex gloss eng> == ENG_I.

R_jiri:
<> == C_R_VR
<phon under> == jiri
<lex gloss eng> == ENG_sleep_pl.

R_kaafa:
<> == C_R_NW
<phon under> == kaafa
<lex gloss eng> == ENG_animal.

R_karu:
<> == C_R_VR_R
<phon under> == karu
<lex gloss eng> == ENG_cut.

R_kati:
<> == C_R_VR
<phon under> == kati
<lex gloss eng> == ENG_encircle.

R_kesebu:
<> == C_R_NW
<phon under> == kesebu
<lex gloss eng> == ENG_knife.

R_kobibi:
<> == "R_kobibi_1" "R_kobibi_2".

R_kobibi_1:
<> == C_R_Adj
<phon under> == kobibi
<lex gloss eng> == ENG_crooked.

R_kobibi_2:
<> == C_R_Adj_VR
<phon under> == kobibi
<lex gloss eng> == ENG_become_crooked.

R_korai:
<> == C_R_VR
<phon under> == korai
<lex gloss eng> == ENG_tear.

R_kou:
<> == C_R_VR_R
<phon under> == kou
```

```
<lex gloss eng> == ENG_break_in_two.

R_kui:
<> == "R_kui_1" "R_kui_2".

R_kui_1:
<> == C_R_Adj
<phon under> == kui
<lex gloss eng> == ENG_wild.

R_kui_2:
<> == C_R_Adj_VR
<phon under> == kui
<lex gloss eng> == ENG_become_wild.

R_kuru:
<> == C_R_VR
<phon under> == kuru
<lex gloss eng> == ENG_look_for.

R_la:
<> == C_R_Prep
<phon under> == la
<lex gloss eng> == ENG_at.

R_lala:
<> == "R_lala_1" "R_lala_2".

R_lala_1:
<> == C_R_NW
<phon under> == lala
<lex gloss eng> == ENG_light
<lex under> == lala_1.

R_lala_2:
<> == C_R_VR
<phon under> == lala
<lex gloss eng> == ENG_shine
<lex under> == lala_2.

R_le:
<> == C_R_VR
<phon under> == le
<lex gloss eng> == ENG_speak.

R_mai:
<> == C_R_VR
<phon under> == mai
<lex gloss eng> == ENG_come_up
<lex form> == DEFAULT GLIDE.

R_matane:
<> == C_R_NW
<phon under> == matane
<lex gloss eng> == ENG_village.

R_me:
```

```
<> == C_R_VR_MC1
<phon under> == me
<lex gloss eng> == ENG_come_down.

R_megebu:
<> == C_R_NW
<phon under> == megebu
<lex gloss eng> == ENG_stone.

R_menekele:
<> == "R_menekele_1" "R_menekele_2".

R_menekele_1:
<> == C_R_Adj
<phon under> == menekele
<lex gloss eng> == ENG_small
<lex form> == DEFAULT ALLO2
<allo form allo2> == melekene.

R_menekele_2:
<> == C_R_Adj_VR
<phon under> == menekele
<lex gloss eng> == ENG_become_small
<lex form> == DEFAULT ALLO2
<allo form allo2> == melekene
<allo mec allo2> == "MEC_01".

R_mini:
<> == C_R_VR_MC1
<phon under> == mini
<lex gloss eng> == ENG_go_down.

R_moge:
<> == C_R_VR
<phon under> == moge
<lex gloss eng> == ENG_turn_back_on.

R_mutu:
<> == C_R_PNR
<phon under> == mutu
<lex gloss eng> == ENG_nose.

R_nage:
<> == C_R_RflPro
<phon under> == nage
<lex gloss eng> == ENG_theirselves_plural.

R_name:
<> == C_R_PosPro
<phon under> == name
<lex gloss eng> == ENG_your_sing.

R_nare:
<> == C_R_RflPro
<phon under> == nare
<lex gloss eng> == ENG_theirselves_dual.

```

```
R_nari:
<> == C_R_VR_MC1
<phon under> == nari
<lex gloss eng> == ENG_hit_O2s.

R_narige:
<> == C_R_VR_MC1
<phon under> == narige
<lex gloss eng> == ENG_hit_O3p.

R_narire:
<> == C_R_VR_MC1
<phon under> == narire
<lex gloss eng> == ENG_hit_O3d.

R_ne:
<> == C_R_PerPro
<phon under> == ne
<lex gloss eng> == ENG_you_sing.

R_nebere:
<> == C_R_PosPro
<phon under> == nebere
<lex gloss eng> == ENG_their_plural.

R_neire:
<> == C_R_PosPro
<phon under> == neire
<lex gloss eng> == ENG_their_dual.

R_nere:
<> == C_R_PerPro
<phon under> == nere
<lex gloss eng> == ENG_they.

R_nerige:
<> == C_R_VR_MC1
<phon under> == nerige
<lex gloss eng> == ENG_see_O3p.

R_nerire:
<> == C_R_VR_MC1
<phon under> == nerire
<lex gloss eng> == ENG_see_O3d.

R_neriye:
<> == C_R_VR_MC1
<phon under> == neriye
<lex gloss eng> == ENG_see_O2s.

R_ngalenga:
<> == "R_ngalenga_1" "R_ngalenga_2".

R_ngalenga_1:
<> == C_R_Adj
<phon under> == ngalenga
<lex gloss eng> == ENG_true.
```

```

R_ngalenga_2:
<> == C_R_Adj_VR
<phon under> == ngalenga
<lex gloss eng> == ENG_become_true.

R_ngare:
<> == C_R_Prep
<phon under> == ngare
<lex gloss eng> == ENG_with_dual.

R_nigi:
<> == C_R_VR_MC1
<phon under> == nigi
<lex gloss eng> == ENG_give_O3p.

R_nina:
<> == C_R_PNR
<phon under> == nina
<lex gloss eng> == ENG_thought.

R_niri:
<> == C_R_VR_MC1
<phon under> == niri
<lex gloss eng> == ENG_give_O3d.

R_no:
<> == C_R_PerPro
<phon under> == no
<lex gloss eng> == ENG_he/she/it.

R_nogo:
<> == C_R_RflPro
<phon under> == nogo
<lex gloss eng> == ENG_himself/herself/itself.

R_nomo:
<> == C_R_PosPro
<phon under> == nomo
<lex gloss eng> == ENG_his/her/its.

R_nuu:
<> == C_R_VR
<phon under> == nuu
<lex gloss eng> == ENG_shove.

R_nyi:
<> == C_R_VR_MC1
<phon under> == nyi
<lex gloss eng> == ENG_consume.

R_ogola:
<> == C_R_NW
<phon under> == ogola
<lex gloss eng> == ENG_cloth.

R_ono:
```

```
<> == C_R_VR
<phon under> == ono
<lex gloss eng> == ENG_do_thus.

R_onou:
<> == C_R_Adj
<phon under> == onou
<lex gloss eng> == ENG_like_that.

R_oofa:
<> == C_R_VR
<phon under> == oofa
<lex gloss eng> == ENG_watch_over.

R_oojo:
<> == C_R_VR
<phon under> == oojo
<lex gloss eng> == ENG_follow.

R_oto:
<> == C_R_VR
<phon under> == oto
<lex gloss eng> == ENG_stand.

R_ou:
<> == "R_ou_1" "R_ou_2".

R_ou_1:
<> == C_R_Adj
<phon under> == ou
<lex gloss eng> == ENG_heavy.

R_ou_2:
<> == C_R_Adj_VR
<phon under> == ou
<lex gloss eng> == ENG_become_heavy.

R_owo:
<> == C_R_PNR
<phon under> == owo
<lex gloss eng> == ENG_hand.

R_soo:
<> == C_R_VR
<phon under> == soo
<lex gloss eng> == ENG_chase.

R_taate:
<> == C_R_IP
<phon under> == taate
<lex gloss eng> == ENG_what.

R_tafa:
<> == C_R_VR_MC1
<phon under> == tafa
<lex gloss eng> == ENG_put.

```

```
R_tage:
<> == C_R_RflPro
<phon under> == tage
<lex gloss eng> == ENG_yourselves_plural.

R_tare:
<> == C_R_RflPro
<phon under> == tare
<lex gloss eng> == ENG_yourselves_dual.

R_tarige:
<> == C_R_VR_MC1
<phon under> == tarige
<lex gloss eng> == ENG_hit_O2p.

R_tarire:
<> == C_R_VR_MC1
<phon under> == tarire
<lex gloss eng> == ENG_hit_O2d.

R_tau:
<> == C_R_VR_R
<phon under> == tau
<lex gloss eng> == ENG_plant.

R_te:
<> == C_R_VR_MC1
<phon under> == te
<lex gloss eng> == ENG_get/take.

R_tebere:
<> == C_R_PosPro
<phon under> == tebere
<lex gloss eng> == ENG_your_plural.

R_teire:
<> == C_R_PosPro
<phon under> == teire
<lex gloss eng> == ENG_your_dual.

R_tere:
<> == C_R_PerPro
<phon under> == tere
<lex gloss eng> == ENG_you_plural.

R_terige:
<> == C_R_VR_MC1
<phon under> == terige
<lex gloss eng> == ENG_see_O2p.

R_terire:
<> == C_R_VR_MC1
<phon under> == terire
<lex gloss eng> == ENG_see_O2d.

R_tigi:
<> == C_R_VR_MC1
```

```
<phon under> == tigi
<lex gloss eng> == ENG_give_O2p.

R_tigini:
<> == C_R_Adj
<phon under> == tigini
<lex gloss eng> == ENG_straight.

R_tigini_2:
<> == C_R_Adj_VR
<phon under> == tigini
<lex gloss eng> == ENG_become_straight.

R_tiri:
<> == C_R_VR_MC1
<phon under> == tiri
<lex gloss eng> == ENG_give_O2d.

R_tu:
<> == C_R_VR
<phon under> == tu
<lex gloss eng> == ENG_give_O3s.

R_tuN:
<> == C_R_VR
<phon under> == tuN
<lex gloss eng> == ENG_poke
<lex form> == NASAL.

R_tutu:
<> == "R_tutu_1" "R_tutu_2".

R_tutu_1:
<> == C_R_Adj
<phon under> == tutu
<lex gloss eng> == ENG_short.

R_tutu_2:
<> == C_R_Adj_VR
<phon under> == tutu
<lex gloss eng> == ENG_become_short.

R_ume:
<> == C_R_VR_MC1
<phon under> == ume
<lex gloss eng> == ENG_die.

R_umuge:
<> == C_R_VR_MC1
<phon under> == umuge
<lex gloss eng> == ENG_fear.

R_ure:
<> == C_R_VR_MC1
<phon under> == ure
<lex gloss eng> == ENG_see_O3s.

```

```
R_usu:
<> == C_R_VR
<phon under> == usu
<lex gloss eng> == ENG_finish.

R_uulu:
<> == C_R_VR
<phon under> == uulu
<lex gloss eng> == ENG_remove.

R_uuwa:
<> == "R_uuwa_1" "R_uuwa_2".

R_uuwa_1:
<> == C_R_NomNeg
<phon under> == uuwa
<lex gloss eng> == ENG_N_not
<lex under> == uuwa_1.

R_uuwa_2:
<> == C_R_Adj_VR
<phon under> == uuwa
<lex gloss eng> == ENG_become_nothing.

R_uwau:
<> == C_R_Prep
<phon under> == uwau
<lex gloss eng> == ENG_without.

R_wa:
<> == C_R_Deictic
<phon under> == wa
<lex gloss eng> == ENG_that.

R_wara:
<> == C_R_VR_MC1
<phon under> == wara
<lex gloss eng> == ENG_fight.

R_wari:
<> == C_R_VR_MC1
<phon under> == wari
<lex gloss eng> == ENG_hit_O3s.

R_wewe:
<> == C_R_NW
<phon under> == wewe
<lex gloss eng> == ENG_dream.

R_yaawa:
<> == C_R_VR
<phon under> == yaawa
<lex gloss eng> == ENG_lie.

R_yafa:
<> == C_R_VR_MC1
<phon under> == yafa
```

```
<lex gloss eng> == ENG_sit.

R_yage:
<> == C_R_RflPro
<phon under> == yage
<lex gloss eng> == ENG_myself.

R_yame:
<> == C_R_PosPro
<phon under> == yame
<lex gloss eng> == ENG_my.

R_yari:
<> == C_R_VR_MC1
<phon under> == yari
<lex gloss eng> == ENG_hit_O1s.

R_yaure:
<> == C_R_VR
<phon under> == yaure
<lex gloss eng> == ENG_yell_to/shout.

R_yerege:
<> == C_R_VR_MC1
<phon under> == yerege
<lex gloss eng> == ENG_write.

R_yeriye:
<> == C_R_VR_MC1
<phon under> == yeriye
<lex gloss eng> == ENG_see_O1s.

S_0:
<> ==       "S_0_Ind_S1d"
            "S_0_Ind_S1p"
            "S_0_Pro_Poss"
            "S_0_1_Ind_S3s"
            "S_0_2_Ind_S3s"
            "S_0_Ind_Tip"
            "S_0_Ind_Tp"
            "S_0_1"
            "S_0_2".

S_0_1:
<> == C_S_DIFF_REF_TEMPORALITY
<phon under> == 0
<lex gloss eng> == ENG_SR_TS.

S_0_1_Ind_S3s:
<> == C_S_IND_SUBJ
<phon under> == 0
<allo sec> == "SEC_11"
<allo mec> == "MEC_11" "MEC_13".

S_0_2:
<> == C_S_SAME_REF_TEMPORALITY
<phon under> == 0
```

```
<lex gloss eng> == ENG_TO.

S_0_2_Ind_S3s:
<> == C_S_IND_SUBJ
<phon under> == 0
<allo sec> == "SEC_10"
<allo mec> == "MEC_07".

S_0_Ind_S1d:
<> == C_S_IND_SUBJ
<phon under> == 0
<allo sec> == "SEC_08"
<allo mec> == "MEC_06".

S_0_Ind_S1p:
<> == C_S_IND_SUBJ
<phon under> == 0
<allo sec> == "SEC_08"
<allo mec> == "MEC_06".

S_0_Ind_Tip:
<> == C_S_IND_TENSE
<lex form> == DEFAULT
<allo mec default> == "MEC_16" "MEC_17"
<allo sec> ==
<phon under> == 0.

S_0_Ind_Tp:
<> == C_S_IND_TENSE
<lex form> == DEFAULT
<allo mec default> == "MEC_22"
<allo sec> ==
<phon under> == 0.

S_0_Pro_Poss:
<> == C_S_PRONOUN_POSS
<lex form> == DEFAULT ALLO2
<phon under> == 0
<allo form allo2> == u
<lex gloss eng> == ENG_P3s.

S_a:
<> ==       "S_a_1"
      "S_a_2"
      "S_a_Ind_Tf"
      "S_a_Ind_Tip"
      "S_a_Ind_Tp"
      "S_a_Ind_Trp".

S_a_1:
<> == C_S_CONTRA_SUBJ
<phon under> == a
<lex gloss eng> == ENG_Contra_S2/3s/d/p.

S_a_2:
<> == C_S_SUBJUNCTIVE_SUBJ_PERSON
<phon under> == a
```

```
<lex gloss eng> == ENG_Sj_S2.

S_a_Ind_Tf:
<> == C_S_IND_TENSE
<phon under> == a
<allo mec> == "MEC_19".

S_a_Ind_Tip:
<> == C_S_IND_TENSE
<phon under> == a
<allo mec> == "MEC_21".

S_a_Ind_Tp:
<> == C_S_IND_TENSE
<phon under> == a
<allo mec> == "MEC_21".

S_a_Ind_Trp:
<> == C_S_IND_TENSE
<phon under> == a
<allo mec> == "MEC_21".

S_boro:
<> == C_S_VERBAL_SUFFIX
<phon under> == boro
<lex gloss eng> == ENG_completely.

S_d:
<> ==       "S_d_Ind_S3d"
      "S_d_Ind_S3p".

S_d_Ind_S3d:
<> == C_S_IND_SUBJ
<phon under> == d
<allo sec> == "SEC_08"
<allo mec> == "MEC_06".

S_d_Ind_S3p:
<> == C_S_IND_SUBJ
<phon under> == d
<allo sec> == "SEC_08"
<allo mec> == "MEC_06".

S_de:
<> ==       "S_de_Ind_S1p"
            "S_de_Ind_S2p"
            "S_de_vAux"
            "S_de_1"
            "S_de_1_Ind_S3p"
            "S_de_2_Ind_S3p".

S_de_1:
<> == C_S_VERBAL_SUFFIX
<phon under> == de
<lex gloss eng> == ENG_well.

S_de_1_Ind_S3p:
```

```
<> == C_S_IND_SUBJ
<phon under> == de
<allo mec> == "MEC_12".

S_de_2_Ind_S3p:
<> == C_S_IND_SUBJ
<phon under> == de
<allo sec> == "SEC_10"
<allo mec> == "MEC_14".

S_de_Ind_S1p:
<> == C_S_IND_SUBJ
<phon under> == de
<allo sec> == "SEC_09"
<allo mec> == "MEC_07"
<lex gloss eng> == S_Ind_S1p.

S_de_Ind_S2p:
<> == C_S_IND_SUBJ
<phon under> == de
<allo sec> == "SEC_11"
<allo mec> == "MEC_07".

S_de_vAux:
<> == C_S_V_AUX
<phon under> == de
<lex gloss eng> == ENG_vAux_S3p.

S_dere:
<> ==       "S_dere_1_Ind_S3d"
            "S_dere_2_Ind_S3d"
            "S_dere_vAux"
            "S_dere_Ind_S1d"
            "S_dere_Ind_S2d"
            "S_dere_vAux".

S_dere_1_Ind_S3d:
<> == C_S_IND_SUBJ
<phon under> == dere
<allo mec> == "MEC_12".

S_dere_2_Ind_S3d:
<> == C_S_IND_SUBJ
<phon under> == dere
<allo sec> == "SEC_10"
<allo mec> == "MEC_06".

S_dere_Ind_S1d:
<> == C_S_IND_SUBJ
<phon under> == dere
<allo sec> == "SEC_09"
<allo mec> == "MEC_07".

S_dere_Ind_S2d:
<> == C_S_IND_SUBJ
<phon under> == dere
<allo sec> == "SEC_11"
```

```
<allo mec> == "MEC_07".

S_dere_vAux:
<> == C_S_V_AUX
<phon under> == dere
<lex gloss eng> == ENG_vAux_S3d.

S_du:
<> == C_S_SAME_REF_TEMPORALITY
<phon under> == du
<lex gloss eng> == ENG_TS.

S_e:
<> ==       "S_e_Contra"
            "S_e_Ind_Tf"
            "S_e_Ind_Tip"
            "S_e_Ind_Tp"
            "S_e_Ind_Trp"
            "S_e_Inf".

S_e_Contra:
<> == C_S_CONTRA_SUBJ
<phon under> == e
<lex gloss eng> == ENG_Contra_S1s.

S_e_Ind_Tf:
<> == C_S_IND_TENSE
<phon under> == e
<allo mec> == "MEC_18".

S_e_Ind_Tip:
<> == C_S_IND_TENSE
<phon under> == e
<allo mec> == "MEC_16".

S_e_Ind_Tp:
<> == C_S_IND_TENSE
<phon under> == e
<allo mec> == "MEC_16".

S_e_Ind_Trp:
<> == C_S_IND_TENSE
<phon under> == e
<allo mec> == "MEC_16".

S_e_Inf:
<> == C_S_INF
<phon under> == e
<lex form> == DEFAULT EPENTHETIC_W
<lex order class> == 80
<lex cat> == 'VR/Inf'
<allo sec default> == "SEC_03"
<lex gloss eng> == ENG_Inf.

S_ede:
<> == C_S_CONTRA_SUBJ
<phon under> == ede
```

```
<lex gloss eng> == ENG_Contra_S1p.

S_edere:
<> == C_S_CONTRA_SUBJ
<phon under> == edere
<lex gloss eng> == ENG_Contra_S1d.

S_fe:
<> ==       "S_fe_1"
            "S_fe_2".

S_fe_1:
<> == C_SUFFIX
<phon under> == fe
<lex order class> == 100
<lex cat> == 'NW/NW Deictic/Deictic Adj/Adj'
<lex gloss eng> == ENG_like1.

S_fe_2:
<> == C_S_PRONOUN_POSS
<phon under> == fe
<lex gloss eng> == ENG_P1s.

S_fu:
<> == C_S_BENEFACTIVE
<phon under> == fu
<lex gloss eng> == ENG_B3s.

S_g:
<> == "S_g_1" "S_g_2".

S_g_1:
<> == C_S_VERBAL_SUFFIX
<phon under> == g
<lex gloss eng> == ENG_Hab
<lex order class> == 70
<lex cat> == 'VR/SFV VR/Nom'.

S_g_2:
<> == C_S_DIFF_REF_TEMPORALITY
<phon under> == g
<lex gloss eng> == ENG_SR_TO.

S_ga:
<> == C_S_SUBJUNCTIVE
<phon under> == ga
<lex gloss eng> == ENG_Prediction
<lex order class> == 100
<lex cat> == 'VSjS/VSjS'.

S_gana:
<> == C_S_PLURAL
<phon under> == gana
<lex gloss eng> == ENG_p1.

S_ge:
<> == "S_ge_1" "S_ge_2".
```

```

S_ge_1:
<> == C_S_CONTRA
<phon under> == ge
<lex gloss eng> == ENG_Contra
<lex form> == DEFAULT
<lex order class> == 100
<lex cat> == 'Contra/Contra'
<allo mec> == "MEC_25".

S_ge_2:
<> == C_S_PRONOUN_POSS
<phon under> == ge
<lex gloss eng> == ENG_P1p.

S_go:
<> == C_S_IND_SUBJ
<phon under> == go
<allo sec> == "SEC_10"
<allo mec> == "MEC_07".

S_gu:
<> ==       "S_gu_Subj"
            "S_gu_Inf".

S_gu_Subj:
<> == C_S_SUBJUNCTIVE_SUBJ_NUMBER
<phon under> == gu
<lex gloss eng> == ENG_Sj_Sp.

S_gu_Inf:
<> == C_S_INF_SUBJ
<phon under> == gu
<lex gloss eng> == ENG_Inf_Sp.

S_ha:
<> == "S_ha_1" "S_ha_2".

S_ha_1:
<> == C_S_ASPECT
<phon under> == ha
<lex gloss eng> == ENG_perf
<lex order class> == 100
<lex cat> == 'SFV/SFVW'.

S_ha_2:
<> == C_SUFFIX
<phon under> == ha
<lex gloss eng> == ENG_pertaining_to
<lex cat> == 'Adj/Adj NW/NW Deictic/Deictic PosPro/PosPro'.

S_hagi:
<> == C_S_BENEFACTIVE_MC1
<phon under> == hagi
<lex gloss eng> == ENG_B1p.

S_hai:
```

```
<> == C_S_BENEFACTIVE
<phon under> == hai
<lex gloss eng> == ENG_B1s
<lex form> == DEFAULT GLIDE.

S_hani:
<> == C_S_BENEFACTIVE_MC1
<phon under> == hani
<lex gloss eng> == ENG_B2s.

S_hari:
<> == C_S_BENEFACTIVE_MC1
<phon under> == hari
<lex gloss eng> == ENG_B1d.

S_he:
<> ==       "S_he_Ind_S1s"
            "S_he_Ind_S2s".

S_he_Ind_S1s:
<> == C_S_IND_SUBJ
<phon under> == he
<allo sec> == "SEC_09"
<allo mec> == "MEC_07".

S_he_Ind_S2s:
<> == C_S_IND_SUBJ
<phon under> == he
<allo sec> == "SEC_11"
<allo mec> == "MEC_07".

S_higi:
<> == C_S_OBJECT_MC1
<phon under> == higi
<lex gloss eng> == ENG_O1p.

S_hii:
<> == C_S_OBJECT
<phon under> == hii
<lex gloss eng> == ENG_O1s
<lex form> == ELISION GLIDE.

S_hini:
<> == C_S_OBJECT_MC1
<phon under> == hini
<lex gloss eng> == ENG_O2s.

S_hiri:
<> == C_S_OBJECT_MC1
<phon under> == hiri
<lex gloss eng> == ENG_O1d.

S_i:
<> == "S_i_Ind_S3s" "S_i_Inf".

S_i_Ind_S3s:
<> == C_S_IND_SUBJ
```

```
<phon under> == i
<allo sec> == "SEC_11"
<allo mec> == "MEC_09".

S_i_Inf:
<> == C_S_INF_SUBJ
<phon under> == i
<lex gloss eng> == ENG_Inf_Ss.

S_Ind_S1d:
<> == C_S_IND_SUBJ
<lex form> ==     "S_0_Ind_S1d"
                  "S_dere_Ind_S1d"
                  "S_re_Ind_S1d"
                  "S_yare_1_Ind_S1d"
                  "S_yare_2_Ind_S1d"
<phon under> == yare
<lex gloss eng> == ENG_Ind_S1d.

S_Ind_S1p:
<> == C_S_IND_SUBJ
<lex form> ==     "S_0_Ind_S1p"
                  "S_de_Ind_S1p"
                  "S_ya_1_Ind_S1p"
                  "S_ya_2_Ind_S1p"
                  "S_ye_Ind_S1p"
<phon under> == ya
<lex gloss eng> == ENG_Ind_S1p.

S_Ind_S1s:
<> == C_S_IND_SUBJ
<lex form> ==     "S_he_Ind_S1s"
                  "S_n_Ind_S1s"
                  "S_na_Ind_S1s"
                  "S_ne_1_Ind_S1s"
                  "S_ne_2_Ind_S1s"
                  "S_ni_1_Ind_S1s"
                  "S_ni_2_Ind_S1s"
<phon under> == ni
<lex gloss eng> == ENG_Ind_S1s.

S_Ind_S2d:
<> == C_S_IND_SUBJ
<lex form> == "S_dere_Ind_S2d" "S_re_Ind_S2d" "S_yare_Ind_S2d"
<phon under> == yare
<lex gloss eng> == ENG_Ind_S2d.

S_Ind_S2p:
<> == C_S_IND_SUBJ
<lex form> == "S_de_Ind_S2p" "S_ya_Ind_S2p" "S_ye_Ind_S2p"
<phon under> == ya
<lex gloss eng> == ENG_Ind_S2p.

S_Ind_S2s:
<> == C_S_IND_SUBJ
<lex form> == "S_he_Ind_S2s" "S_na_Ind_S2s" "S_ne_1_Ind_S2s"
                          "S_ne_2_Ind_S2s" "S_ni_Ind_S2s"
```

```
         <phon under> == ni
         <lex gloss eng> == ENG_Ind_S2s.

         S_Ind_S3d:
         <> == C_S_IND_SUBJ
         <lex form> == "S_d_Ind_S3d" "S_dere_1_Ind_S3d"
                                   "S_dere_2_Ind_S3d" "S_re_Ind_S3d"
"S_ro"
         <phon under> == dere
         <lex gloss eng> == ENG_Ind_S3d.

         S_Ind_S3p:
         <> == C_S_IND_SUBJ
         <lex form> == "S_d_Ind_S3p" "S_de_1_Ind_S3p" "S_de_2_Ind_S3p"
                                   "S_go" "S_ne_Ind_S3p"
         <phon under> == de
         <lex gloss eng> == ENG_Ind_S3p.

         S_Ind_S3s:
         <> == C_S_IND_SUBJ
         <lex form> == "S_0_1_Ind_S3s" "S_0_2_Ind_S3s" "S_i_Ind_S3s"
                                   "S_n_Ind_S3s" "S_na_1_Ind_S3s"
"S_ta"
                                   "S_na_2_Ind_S3s"
         <phon under> == na
         <lex gloss eng> == ENG_Ind_S3s.

         S_Ind_Tf:
         <> == C_S_IND_TENSE
         <lex form> == "S_e_Ind_Tf" "S_a_Ind_Tf" "S_o_Ind_Tf"
         <phon under> == o
         <lex gloss eng> == ENG_Ind_Tf.

         S_Ind_Tip:
         <> == C_S_IND_TENSE
         <lex form> == "S_e_Ind_Tip" "S_a_Ind_Tip" "S_0_Ind_Tip"
         <phon under> == a
         <lex gloss eng> == ENG_Ind_Tip.

         S_Ind_Tp:
         <> == C_S_IND_TENSE
         <lex form> == "S_0_Ind_Tp" "S_e_Ind_Tp" "S_a_Ind_Tp"
         <phon under> == 0
         <lex gloss eng> == ENG_Ind_Tp.

         S_Ind_Trp:
         <> == C_S_IND_TENSE
         <lex form> == "S_e_Ind_Trp" "S_a_Ind_Trp"
         <phon under> == e
         <lex gloss eng> == ENG_Ind_Trp.

         S_mu:
         <> == C_S_VERBAL_SUFFIX
         <phon under> == mu
         <lex gloss eng> == ENG_causative
         <lex order class> == 10.

```

```
S_n:
<> == "S_n_Ind_S1s" "S_n_Ind_S3s".

S_n_Ind_S1s:
<> == C_S_IND_SUBJ
<phon under> == n
<allo sec> == "SEC_08"
<allo mec> == "MEC_06".

S_n_Ind_S3s:
<> == C_S_IND_SUBJ
<phon under> == n
<allo sec> == "SEC_08"
<allo mec> == "MEC_06".

S_na:
<> == "S_na_1_Ind_S3s" "S_na_2" "S_na_2_Ind_S3s"
                  "S_na_Ind_S1s" "S_na_Ind_S2s".

S_na_1_Ind_S3s:
<> == C_S_IND_SUBJ
<phon under> == na
<allo mec> == "MEC_12".

S_na_2:
<> == C_S_V_AUX
<phon under> == na
<lex gloss eng> == ENG_vAux_S3s.

S_na_2_Ind_S3s:
<> == C_S_IND_SUBJ
<phon under> == na
<allo sec> == "SEC_10"
<allo mec> == "MEC_06".

S_na_Ind_S1s:
<> == C_S_IND_SUBJ
<phon under> == na
<allo mec> == "MEC_11".

S_na_Ind_S2s:
<> == C_S_IND_SUBJ
<phon under> == na
<allo mec> == "MEC_11".

S_nade:
<> == C_S_V_AUX
<phon under> == nade
<lex gloss eng> == ENG_vAux_Tf_S2p.

S_nadere:
<> == C_S_V_AUX
<phon under> == nadere
<lex gloss eng> == ENG_vAux_Tf_S2d.

S_nagi:
<> == C_S_BENEFACTIVE_MC1
```

```
<phon under> == nagi
<lex gloss eng> == ENG_B3p.

S_nahe:
<> == C_S_V_AUX
<phon under> == nahe
<lex gloss eng> == ENG_vAux_Tf_S2s.

S_nari:
<> == C_S_BENEFACTIVE_MC1
<phon under> == nari
<lex gloss eng> == ENG_B3d.

S_ne:
<> == "S_ne_1_Ind_S1s" "S_ne_1_Ind_S2s" "S_ne_2"
                  "S_ne_2_Ind_S2s" "S_ne_2_Ind_S2s" "S_ne_Ind_S3p"
                  "S_ne_vAux".

S_ne_1_Ind_S1s:
<> == C_S_IND_SUBJ
<phon under> == ne
<allo mec> == "MEC_08".

S_ne_1_Ind_S2s:
<> == C_S_IND_SUBJ
<phon under> == ne
<allo mec> == "MEC_13".

S_ne_2:
<> == C_S_PRONOUN_POSS
<phon under> == ne
<lex gloss eng> == ENG_P2s
<lex form> == DEFAULT ALLO2
<allo form allo2> == te.

S_ne_2_Ind_S1s:
<> == C_S_IND_SUBJ
<phon under> == ne
<allo sec> == "SEC_09"
<allo mec> == "MEC_08" "MEC_09".

S_ne_2_Ind_S2s:
<> == C_S_IND_SUBJ
<phon under> == ne
<allo sec> == "SEC_09"
<allo mec> == "MEC_08" "MEC_09".

S_ne_Ind_S3p:
<> == C_S_IND_SUBJ
<phon under> == ne
<allo sec> == "SEC_11"
<allo mec> == "MEC_11" "MEC_13" "MEC_09".

S_ne_vAux:
<> == C_S_V_AUX
<phon under> == ne
```

```
<lex gloss eng> == ENG_vAux_Tnf_S2s.

S_nege:
<> == C_S_PRONOUN_POSS
<phon under> == nege
<lex gloss eng> == ENG_P3p.

S_nere:
<> == C_S_PRONOUN_POSS
<phon under> == nere
<lex gloss eng> == ENG_P3d.

S_neu:
<> == C_SUFFIX
<phon under> == neu
<lex order class> == 100
<lex cat> == 'Deictic/Deictic'
<lex gloss eng> == ENG_like2
<allo mec> == "MEC_24".

S_nga:
<> == "S_nga_1" "S_nga_2" "S_nga_3".

S_nga_1:
<> == C_SUFFIX
<phon under> == nga
<lex cat> == 'NW/NW RflPro/RflPro PN/PN Adj/Adj PosPro/PosPro
                         PerPro/PerPro Prep/Prep
Deictic/Deictic VR/VR'
<lex gloss eng> == ENG_just/only.

S_nga_2:
<> == C_S_SUBJUNCTIVE
<phon under> == nga
<lex gloss eng> == ENG_certainty
<lex order class> == 110
<lex cat> == 'VSjS/VSjS'.

S_nga_3:
<> == C_S_VERBAL_SUFFIX
<phon under> == nga
<lex gloss eng> == ENG_SR
<lex order class> == 110
<lex cat> == 'SMV/SMVW Contra/SMVW' .

S_ni:
<> == "S_ni_vAux" "S_ni_1_Ind_S1s" "S_ni_2_Ind_S1s"
                  "S_ni_Ind_S2s".

S_ni_vAux:
<> == C_S_V_AUX
<phon under> == ni
<lex gloss eng> == ENG_vAux_S1s.

S_ni_1_Ind_S1s:
<> == C_S_IND_SUBJ
<phon under> == ni
```

```
<allo mec> == "MEC_12" "MEC_08".

S_ni_2_Ind_S1s:
<> == C_S_IND_SUBJ
<phon under> == ni
<allo sec> == "SEC_10"
<allo mec> == "MEC_06".

S_ni_Ind_S2s:
<> == C_S_IND_SUBJ
<phon under> == ni
<allo mec> == "MEC_12" "MEC_08".

S_nigi:
<> == C_S_OBJECT_MC1
<phon under> == nigi
<lex gloss eng> == ENG_O3p.

S_niri:
<> == C_S_OBJECT_MC1
<phon under> == niri
<lex gloss eng> == ENG_O3d.

S_o:
<> == "S_o_Subj" "S_o_Ind_Tf".

S_o_Subj:
<> == C_S_SUBJUNCTIVE_SUBJ_PERSON
<phon under> == o
<lex gloss eng> == ENG_Sj_S1/3.

S_o_Ind_Tf:
<> == C_S_IND_TENSE
<phon under> == o
<allo mec> == "MEC_20".

S_ou:
<> == "S_ou_Attrib" "S_ou_Nom".

S_ou_Attrib:
<> == C_S_ATTRIB
<phon under> == ou
<lex gloss eng> == ENG_Attrib.

S_ou_Nom:
<> == C_SUFFIX
<phon under> == ou
<lex form> == DEFAULT EPENTHETIC_W
<lex order class> == 110
<lex cat> == 'VR/Nom Nom/Nom'
<allo sec default> == "SEC_03"
<lex gloss eng> == ENG_Nom.

S_re:
<> == "S_re_Ind_S1d" "S_re_Ind_S2d" "S_re_2" "S_re_Ind_S3d".

S_re_2:
```

```
<> == C_S_PRONOUN_POSS
<phon under> == re
<lex gloss eng> == ENG_P1d.

S_re_Ind_S1d:
<> == C_S_IND_SUBJ
<phon under> == re
<allo sec> == "SEC_09"
<allo mec> == "MEC_08" "MEC_09".

S_re_Ind_S2d:
<> == C_S_IND_SUBJ
<phon under> == re
<allo sec> == "SEC_09"
<allo mec> == "MEC_08" "MEC_09".

S_re_Ind_S3d:
<> == C_S_IND_SUBJ
<phon under> == re
<allo sec> == "SEC_11"
<allo mec> == "MEC_11" "MEC_13" "MEC_09".

S_ro:
<> == C_S_IND_SUBJ
<phon under> == ro
<allo sec> == "SEC_10"
<allo mec> == "MEC_07".

S_ru:
<> == "S_ru_1" "S_ru_2" "S_ru_Inf".

S_ru_1:
<> == C_S_PLURAL
<phon under> == ru
<lex gloss eng> == ENG_p2.

S_ru_2:
<> == C_S_SUBJUNCTIVE_SUBJ_NUMBER
<phon under> == ru
<lex gloss eng> == ENG_Sj_Sd.

S_ru_Inf:
<> == C_S_INF_SUBJ
<phon under> == ru
<lex gloss eng> == ENG_Inf_Sd.

S_ta:
<> == C_S_IND_SUBJ
<phon under> == ta
<allo mec> == "MEC_15".

S_tagi:
<> == C_S_BENEFACTIVE_MC1
<phon under> == tagi
<lex gloss eng> == ENG_B2p.

S_tari:
```

```
<> == C_S_BENEFACTIVE_MC1
<phon under> == tari
<lex gloss eng> == ENG_B2d.

S_tege:
<> == C_S_PRONOUN_POSS
<phon under> == tege
<lex gloss eng> == ENG_P2p.

S_tere:
<> == C_S_PRONOUN_POSS
<phon under> == tere
<lex gloss eng> == ENG_P2d.

S_tigi:
<> == C_S_OBJECT_MC1
<phon under> == tigi
<lex gloss eng> == ENG_O2p.

S_tiri:
<> == C_S_OBJECT_MC1
<phon under> == tiri
<lex gloss eng> == ENG_O2d.

S_tu:
<> == C_S_OBJECT
<phon under> == tu
<lex gloss eng> == ENG_O3s.

S_u:
<> == C_S_SUBJUNCTIVE_SUBJ_NUMBER
<phon under> == u
<lex gloss eng> == ENG_Sj_Ss.

S_ya:
<> == "S_ya_1_Ind_S1p" "S_ya_2_Ind_S1p" "S_ya_Ind_S2p"
                  "S_ya_Manner" "S_ya_1_vAux" "S_ya_2_vAux".

S_ya_1_Ind_S1p:
<> == C_S_IND_SUBJ
<phon under> == ya
<allo mec> == "MEC_10".

S_ya_1_vAux:
<> == C_S_V_AUX
<phon under> == ya
<lex gloss eng> == ENG_vAux_S1p.

S_ya_2_Ind_S1p:
<> == C_S_IND_SUBJ
<phon under> == ya
<allo sec> == "SEC_10"
<allo mec> == "MEC_06".

S_ya_2_vAux:
<> == C_S_V_AUX
<phon under> == ya
```

```
<lex gloss eng> == ENG_vAux_Tnf_S2p.

S_ya_Ind_S2p:
<> == C_S_IND_SUBJ
<phon under> == ya
<allo mec> == "MEC_10".

S_ya_Manner:
<> == C_S_VERBAL_SUFFIX
<phon under> == ya
<lex gloss eng> == ENG_quickly_2.

S_yare:
<> == "S_yare_1_Ind_S1d" "S_yare_2_Ind_S1d" "S_yare_1_vAux"
                  "S_yare_2_vAux" "S_yare_Ind_S2d".

S_yare_1_Ind_S1d:
<> == C_S_IND_SUBJ
<phon under> == yare
<allo mec> == "MEC_10".

S_yare_1_vAux:
<> == C_S_V_AUX
<phon under> == yare
<lex gloss eng> == ENG_vAux_S1d.

S_yare_2_Ind_S1d:
<> == C_S_IND_SUBJ
<phon under> == yare
<allo sec> == "SEC_10"
<allo mec> == "MEC_06".

S_yare_2_vAux:
<> == C_S_V_AUX
<phon under> == yare
<lex gloss eng> == ENG_vAux_Tnf_S2d.

S_yare_Ind_S2d:
<> == C_S_IND_SUBJ
<phon under> == yare
<allo mec> == "MEC_10".

S_ye:
<> == "S_ye_Ind_S1p" "S_ye_Ind_S2p".

S_ye_Ind_S1p:
<> == C_S_IND_SUBJ
<phon under> == ye
<allo sec> == "SEC_09"
<allo mec> == "MEC_08" "MEC_09".

S_ye_Ind_S2p:
<> == C_S_IND_SUBJ
<phon under> == ye
<allo sec> == "SEC_09"
<allo mec> == "MEC_08" "MEC_09".

```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A3.3 The Ogea DATR Query File

A3.3.1 Description

DATR theory (lexicon) files contain sets of nodes. Queries against a theory file may be made interactively from a DATR command line, or alternatively, may be run in batch mode. In batch mode, a query file may be used so the user does not have to manually type in each query. The Ogea query file listing in this appendix is used to generate AMPLE entries from the Ogea DATR theory file. Each entry in the query file has a corresponding entry in the original Ogea AMPLE root or suffix database file. The path `<amp lex mor name>` is used to direct the query to invoke the AMPLE interface and to use the morph name as the value for the record marker.

A3.3.2 File Listing

Filename: ogea.qry

Note: line numbers added for reference only. Not in actual file.

```
(6360)      ENG_able:<amp lex morname>
(6361)      ENG_allow/leave:<amp lex morname>
(6362)      ENG_always:<amp lex morname>
(6363)      ENG_animal:<amp lex morname>
(6364)      ENG_appear/arrive:<amp lex morname>
(6365)      ENG_ask:<amp lex morname>
(6366)      ENG_at:<amp lex morname>
(6367)      ENG_Attrib:<amp lex morname>
(6368)      ENG_B1d:<amp lex morname>
(6369)      ENG_B1p:<amp lex morname>
(6370)      ENG_B1s:<amp lex morname>
(6371)      ENG_B2d:<amp lex morname>
(6372)      ENG_B2p:<amp lex morname>
(6373)      ENG_B2s:<amp lex morname>
(6374)      ENG_B3d:<amp lex morname>
(6375)      ENG_B3p:<amp lex morname>
```

```
(6376)      ENG_B3s:<amp lex morname>
(6377)      ENG_bad:<amp lex morname>
(6378)      ENG_be:<amp lex morname>
(6379)      ENG_become_bad:<amp lex morname>
(6380)      ENG_become_crooked:<amp lex morname>
(6381)      ENG_become_good:<amp lex morname>
(6382)      ENG_become_heavy:<amp lex morname>
(6383)      ENG_become_large:<amp lex morname>
(6384)      ENG_become_many:<amp lex morname>
(6385)      ENG_become_nothing:<amp lex morname>
(6386)      ENG_become_red:<amp lex morname>
(6387)      ENG_become_short:<amp lex morname>
(6388)      ENG_become_small:<amp lex morname>
(6389)      ENG_become_straight:<amp lex morname>
(6390)      ENG_become_true:<amp lex morname>
(6391)      ENG_become_wild:<amp lex morname>
(6392)      ENG_bound:<amp lex morname>
(6393)      ENG_break_in_two:<amp lex morname>
(6394)      ENG_causative:<amp lex morname>
(6395)      ENG_certainty:<amp lex morname>
(6396)      ENG_chase:<amp lex morname>
(6397)      ENG_cloth:<amp lex morname>
(6398)      ENG_come_down:<amp lex morname>
(6399)      ENG_come_up:<amp lex morname>
(6400)      ENG_completely:<amp lex morname>
(6401)      ENG_connect/join:<amp lex morname>
(6402)      ENG_consume:<amp lex morname>
(6403)      ENG_Contra:<amp lex morname>
(6404)      ENG_Contra_S1d:<amp lex morname>
(6405)      ENG_Contra_S1p:<amp lex morname>
(6406)      ENG_Contra_S1s:<amp lex morname>
(6407)      ENG_Contra_S2/3s/d/p:<amp lex morname>
(6408)      ENG_crooked:<amp lex morname>
(6409)      ENG_cut:<amp lex morname>
(6410)      ENG_deflect:<amp lex morname>
(6411)      ENG_desire:<amp lex morname>
(6412)      ENG_die:<amp lex morname>
(6413)      ENG_different:<amp lex morname>
(6414)      ENG_do_not:<amp lex morname>
(6415)      ENG_do_thus:<amp lex morname>
(6416)      ENG_do_work:<amp lex morname>
(6417)      ENG_dream:<amp lex morname>
(6418)      ENG_encircle:<amp lex morname>
(6419)      ENG_face:<amp lex morname>
(6420)      ENG_fear:<amp lex morname>
(6421)      ENG_fight:<amp lex morname>
(6422)      ENG_fill:<amp lex morname>
(6423)      ENG_finish:<amp lex morname>
(6424)      ENG_fly:<amp lex morname>
(6425)      ENG_follow:<amp lex morname>
(6426)      ENG_gather:<amp lex morname>
(6427)      ENG_get_O1d:<amp lex morname>
(6428)      ENG_get_O1p:<amp lex morname>
(6429)      ENG_get_O1s:<amp lex morname>
(6430)      ENG_get_O2d:<amp lex morname>
(6431)      ENG_get_O2p:<amp lex morname>
(6432)      ENG_get_O2s:<amp lex morname>
```

```
(6433)      ENG_get_O3d:<amp lex morname>
(6434)      ENG_get_O3p:<amp lex morname>
(6435)      ENG_get_O3s:<amp lex morname>
(6436)      ENG_get_up:<amp lex morname>
(6437)      ENG_get/take:<amp lex morname>
(6438)      ENG_give_O1d:<amp lex morname>
(6439)      ENG_give_O1p:<amp lex morname>
(6440)      ENG_give_O1s:<amp lex morname>
(6441)      ENG_give_O2d:<amp lex morname>
(6442)      ENG_give_O2p:<amp lex morname>
(6443)      ENG_give_O2s:<amp lex morname>
(6444)      ENG_give_O3d:<amp lex morname>
(6445)      ENG_give_O3p:<amp lex morname>
(6446)      ENG_give_O3s:<amp lex morname>
(6447)      ENG_go:<amp lex morname>
(6448)      ENG_go_down:<amp lex morname>
(6449)      ENG_good:<amp lex morname>
(6450)      ENG_Hab:<amp lex morname>
(6451)      ENG_hand:<amp lex morname>
(6452)      ENG_he/she/it:<amp lex morname>
(6453)      ENG_hear/understand/obey:<amp lex morname>
(6454)      ENG_heavy:<amp lex morname>
(6455)      ENG_help:<amp lex morname>
(6456)      ENG_himself/herself/itself:<amp lex morname>
(6457)      ENG_his/her/its:<amp lex morname>
(6458)      ENG_hit_O1d:<amp lex morname>
(6459)      ENG_hit_O1p:<amp lex morname>
(6460)      ENG_hit_O1s:<amp lex morname>
(6461)      ENG_hit_O2d:<amp lex morname>
(6462)      ENG_hit_O2p:<amp lex morname>
(6463)      ENG_hit_O2s:<amp lex morname>
(6464)      ENG_hit_O3d:<amp lex morname>
(6465)      ENG_hit_O3p:<amp lex morname>
(6466)      ENG_hit_O3s:<amp lex morname>
(6467)      ENG_I:<amp lex morname>
(6468)      ENG_idle:<amp lex morname>
(6469)      ENG_Ind_S1d:<amp lex morname>
(6470)      ENG_Ind_S1p:<amp lex morname>
(6471)      ENG_Ind_S1s:<amp lex morname>
(6472)      ENG_Ind_S2d:<amp lex morname>
(6473)      ENG_Ind_S2p:<amp lex morname>
(6474)      ENG_Ind_S2s:<amp lex morname>
(6475)      ENG_Ind_S3d:<amp lex morname>
(6476)      ENG_Ind_S3p:<amp lex morname>
(6477)      ENG_Ind_S3s:<amp lex morname>
(6478)      ENG_Ind_Tf:<amp lex morname>
(6479)      ENG_Ind_Tip:<amp lex morname>
(6480)      ENG_Ind_Tp:<amp lex morname>
(6481)      ENG_Ind_Trp:<amp lex morname>
(6482)      ENG_Inf:<amp lex morname>
(6483)      ENG_Inf_Sd:<amp lex morname>
(6484)      ENG_Inf_Sp:<amp lex morname>
(6485)      ENG_Inf_Ss:<amp lex morname>
(6486)      ENG_inside:<amp lex morname>
(6487)      ENG_judge:<amp lex morname>
(6488)      ENG_just/only:<amp lex morname>
(6489)      ENG_knife:<amp lex morname>
```

```
(6490)      ENG_large:<amp lex morname>
(6491)      ENG_leg/foot:<amp lex morname>
(6492)      ENG_leg/foot_3sP:<amp lex morname>
(6493)      ENG_lie:<amp lex morname>
(6494)      ENG_light:<amp lex morname>
(6495)      ENG_like_that:<amp lex morname>
(6496)      ENG_like1:<amp lex morname>
(6497)      ENG_like2:<amp lex morname>
(6498)      ENG_look_for:<amp lex morname>
(6499)      ENG_man:<amp lex morname>
(6500)      ENG_many:<amp lex morname>
(6501)      ENG_my:<amp lex morname>
(6502)      ENG_myself:<amp lex morname>
(6503)      ENG_N_not:<amp lex morname>
(6504)      ENG_Nom:<amp lex morname>
(6505)      ENG_nose:<amp lex morname>
(6506)      ENG_O1d:<amp lex morname>
(6507)      ENG_O1p:<amp lex morname>
(6508)      ENG_O1s:<amp lex morname>
(6509)      ENG_O2d:<amp lex morname>
(6510)      ENG_O2p:<amp lex morname>
(6511)      ENG_O2s:<amp lex morname>
(6512)      ENG_O3d:<amp lex morname>
(6513)      ENG_O3p:<amp lex morname>
(6514)      ENG_O3s:<amp lex morname>
(6515)      ENG_ocean:<amp lex morname>
(6516)      ENG_one/a:<amp lex morname>
(6517)      ENG_our_dual:<amp lex morname>
(6518)      ENG_our_plural:<amp lex morname>
(6519)      ENG_ourselves_dual:<amp lex morname>
(6520)      ENG_ourselves_plural:<amp lex morname>
(6521)      ENG_p1:<amp lex morname>
(6522)      ENG_P1d:<amp lex morname>
(6523)      ENG_P1p:<amp lex morname>
(6524)      ENG_P1s:<amp lex morname>
(6525)      ENG_p2:<amp lex morname>
(6526)      ENG_P2d:<amp lex morname>
(6527)      ENG_P2p:<amp lex morname>
(6528)      ENG_P2s:<amp lex morname>
(6529)      ENG_P3d:<amp lex morname>
(6530)      ENG_P3p:<amp lex morname>
(6531)      ENG_P3s:<amp lex morname>
(6532)      ENG_perf:<amp lex morname>
(6533)      ENG_pertaining_to:<amp lex morname>
(6534)      ENG_pig:<amp lex morname>
(6535)      ENG_place:<amp lex morname>
(6536)      ENG_plant:<amp lex morname>
(6537)      ENG_poke:<amp lex morname>
(6538)      ENG_pour_out:<amp lex morname>
(6539)      ENG_Prediction:<amp lex morname>
(6540)      ENG_prepare:<amp lex morname>
(6541)      ENG_press:<amp lex morname>
(6542)      ENG_put:<amp lex morname>
(6543)      ENG_quickly:<amp lex morname>
(6544)      ENG_quickly_2:<amp lex morname>
(6545)      ENG_red:<amp lex morname>
(6546)      ENG_remove:<amp lex morname>
```

```
(6547)      ENG_run_away:<amp lex morname>
(6548)      ENG_see_O1d:<amp lex morname>
(6549)      ENG_see_O1p:<amp lex morname>
(6550)      ENG_see_O1s:<amp lex morname>
(6551)      ENG_see_O2d:<amp lex morname>
(6552)      ENG_see_O2p:<amp lex morname>
(6553)      ENG_see_O2s:<amp lex morname>
(6554)      ENG_see_O3d:<amp lex morname>
(6555)      ENG_see_O3p:<amp lex morname>
(6556)      ENG_see_O3s:<amp lex morname>
(6557)      ENG_send:<amp lex morname>
(6558)      ENG_shine:<amp lex morname>
(6559)      ENG_short:<amp lex morname>
(6560)      ENG_shove:<amp lex morname>
(6561)      ENG_show:<amp lex morname>
(6562)      ENG_sibling_same_sex_his/her/its:<amp lex morname>
(6563)      ENG_sibling_same_sex_possessive:<amp lex morname>
(6564)      ENG_sibling_same_sex_referential:<amp lex morname>
(6565)      ENG_sit:<amp lex morname>
(6566)      ENG_Sj_S1/3:<amp lex morname>
(6567)      ENG_Sj_S2:<amp lex morname>
(6568)      ENG_Sj_Sd:<amp lex morname>
(6569)      ENG_Sj_Sp:<amp lex morname>
(6570)      ENG_Sj_Ss:<amp lex morname>
(6571)      ENG_sleep_pl:<amp lex morname>
(6572)      ENG_small:<amp lex morname>
(6573)      ENG_smooth:<amp lex morname>
(6574)      ENG_some:<amp lex morname>
(6575)      ENG_speak:<amp lex morname>
(6576)      ENG_SR:<amp lex morname>
(6577)      ENG_SR_TO:<amp lex morname>
(6578)      ENG_SR_TS:<amp lex morname>
(6579)      ENG_stand:<amp lex morname>
(6580)      ENG_stick_to:<amp lex morname>
(6581)      ENG_stone:<amp lex morname>
(6582)      ENG_straight:<amp lex morname>
(6583)      ENG_talk/word:<amp lex morname>
(6584)      ENG_teach:<amp lex morname>
(6585)      ENG_tear:<amp lex morname>
(6586)      ENG_tell:<amp lex morname>
(6587)      ENG_that:<amp lex morname>
(6588)      ENG_their_dual:<amp lex morname>
(6589)      ENG_their_plural:<amp lex morname>
(6590)      ENG_theirselves_dual:<amp lex morname>
(6591)      ENG_theirselves_plural:<amp lex morname>
(6592)      ENG_they:<amp lex morname>
(6593)      ENG_thigh:<amp lex morname>
(6594)      ENG_thing:<amp lex morname>
(6595)      ENG_this:<amp lex morname>
(6596)      ENG_thought:<amp lex morname>
(6597)      ENG_throw:<amp lex morname>
(6598)      ENG_TO:<amp lex morname>
(6599)      ENG_true:<amp lex morname>
(6600)      ENG_TS:<amp lex morname>
(6601)      ENG_turn_back_on:<amp lex morname>
(6602)      ENG_untie:<amp lex morname>
(6603)      ENG_V_not:<amp lex morname>
```

```
(6604)      ENG_vAux_S1d:<amp lex morname>
(6605)      ENG_vAux_S1p:<amp lex morname>
(6606)      ENG_vAux_S1s:<amp lex morname>
(6607)      ENG_vAux_S3d:<amp lex morname>
(6608)      ENG_vAux_S3p:<amp lex morname>
(6609)      ENG_vAux_S3s:<amp lex morname>
(6610)      ENG_vAux_Tf_S2d:<amp lex morname>
(6611)      ENG_vAux_Tf_S2p:<amp lex morname>
(6612)      ENG_vAux_Tf_S2s:<amp lex morname>
(6613)      ENG_vAux_Tnf_S2d:<amp lex morname>
(6614)      ENG_vAux_Tnf_S2p:<amp lex morname>
(6615)      ENG_vAux_Tnf_S2s:<amp lex morname>
(6616)      ENG_vAuxR:<amp lex morname>
(6617)      ENG_very:<amp lex morname>
(6618)      ENG_village:<amp lex morname>
(6619)      ENG_walk:<amp lex morname>
(6620)      ENG_watch:<amp lex morname>
(6621)      ENG_watch_over:<amp lex morname>
(6622)      ENG_we:<amp lex morname>
(6623)      ENG_well:<amp lex morname>
(6624)      ENG_what:<amp lex morname>
(6625)      ENG_wild:<amp lex morname>
(6626)      ENG_with:<amp lex morname>
(6627)      ENG_with_dual:<amp lex morname>
(6628)      ENG_without:<amp lex morname>
(6629)      ENG_woman:<amp lex morname>
(6630)      ENG_work:<amp lex morname>
(6631)      ENG_write:<amp lex morname>
(6632)      ENG_yell_to/shout:<amp lex morname>
(6633)      ENG_you_plural:<amp lex morname>
(6634)      ENG_you_sing:<amp lex morname>
(6635)      ENG_your_dual:<amp lex morname>
(6636)      ENG_your_plural:<amp lex morname>
(6637)      ENG_your_sing:<amp lex morname>
(6638)      ENG_yourselves_dual:<amp lex morname>
(6639)      ENG_yourselves_plural:<amp lex morname>
```

A3.4 Batch File to Execute DATR

A3.4.1 Description

The batch file listed below is a standard DOS batch file. It contains two statements. The first calls the ZDATR program *zdatrtok*. Zdatrtok produces a token file from the Ogea.dtr file and names it Ogea.dtr.tok. The second program called is *zdatrinf* which is the inference program. This program uses both the Ogea.qry file and the Ogea.dtr.tok file as inputs, and and then outputs the results of the query to a file named Ogea.dtr.trc. Ogea.dtr.trc is the Ogea trace file. The trace file is cleaned up by the Perl script DToAMPLE.pl, which outputs the file Ogea.db, which is directly runable by AMPLE.

A3.4.2 File Listing

Filename: go.bat

Note: line numbers added for reference only. Not in actual file.

```
(6640)      zdatrtok -o Ogea.dtr.tok Ogea.dtr
(6641)      zdatrinf -j -u -V -1 -T Ogea.dtr.trc -Q Ogea.qry Ogea.dtr.tok
(6642)      perl DToAMPLE.pl
```

A3.5 The Ogea DATR Trace File

A3.5.1 Description

The result of running the batch file described above is a trace file called *Ogea.dtr.trc*. Each line in the query file *Ogea.qry* results in a corresponding line in the trace file. Note that although the trace file entries are close to AMPLE format, two things must be done before they are usable by AMPLE. First, empty lines and empty fields must be removed. Second, the trailing period must be stripped off. This is handled by a Perl script listed in Appendix Five.

A3.5.2 File Listing

Filename: ogea.dtr.trc

Note: line numbers added for reference only. Not in actual file.

```
(6643)      \g able
(6644)      \ge able
(6645)      \type r
(6646)      \a edo
(6647)      \u edo
(6648)      \mp
(6649)      \o
(6650)      \c Adv
(6651)      \e
(6652)      \mcc
(6653)      \f
(6654)      \loc
(6655)      \_no
(6656)      \co
(6657)      .
```

```
\g allow/leave
\ge allow/leave
\type r
\a awa
\a aw  / _ [V]
\u awa
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g always
\ge always
\type r
\a gaigai
\u gaigai
\mp
\o
\c Adv
\e
\mcc
\f
\loc
\_no
\co

.
\g animal
\ge animal
\type r
\a kaafa
\u kaafa
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g appear/arrive
\ge appear/arrive
\type r
\a heige
\a heig  / _ [V]
\u heige
\mp MC1
\o
```

```
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g ask
\ge ask
\type r
\a isoki
\u isoki
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g at
\ge at
\type r
\a la
\u la
\mp
\o
\c Prep
\e
\mcc
\f
\loc
\_no
\co

.
\g Attrib
\ge Attributive marker
\type s
\a ou  / [C] _
\a wou  / [V] _
\u ou
\mp
\o 110
\c Adj/Atr
\e
\mcc
\f
\loc
\_no
\co

```

```
.
\g B1d
\ge for us (dual)
\type s
\a hari
\a har  / _ [V]
\u hari
\mp MC1
\o 60
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g B1p
\ge for us (plural)
\type s
\a hagi
\a hag  / _ [V]
\u hagi
\mp MC1
\o 60
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g B1s
\ge for me
\type s
\a hai
\a hay  / _ [V]
\u hai
\mp MC2
\o 60
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g B2d
\ge for you (dual)
\type s
\a tari
\a tar  / _ [V]
```

```
(6829)      \u tari
(6830)      \mp MC1
(6831)      \o 60
(6832)      \c VR/VR
(6833)      \e
(6834)      \mcc
(6835)      \f
(6836)      \loc
(6837)      \_no
(6838)      \co
(6839)
(6840)      .
(6841)      \g B2p
(6842)      \ge for you (plural)
(6843)      \type s
(6844)      \a tagi
(6845)      \a tag  / _ [V]
(6846)      \u tagi
(6847)      \mp MC1
(6848)      \o 60
(6849)      \c VR/VR
(6850)      \e
(6851)      \mcc
(6852)      \f
(6853)      \loc
(6854)      \_no
(6855)      \co
(6856)
(6857)      .
(6858)      \g B2s
(6859)      \ge for you (singular)
(6860)      \type s
(6861)      \a hani
(6862)      \a han  / _ [V]
(6863)      \u hani
(6864)      \mp MC1
(6865)      \o 60
(6866)      \c VR/VR
(6867)      \e
(6868)      \mcc
(6869)      \f
(6870)      \loc
(6871)      \_no
(6872)      \co
(6873)
(6874)      .
(6875)      \g B3d
(6876)      \ge for them (dual)
(6877)      \type s
(6878)      \a nari
(6879)      \a nar  / _ [V]
(6880)      \u nari
(6881)      \mp MC1
(6882)      \o 60
(6883)      \c VR/VR
(6884)      \e
(6885)      \mcc
```

```
\f
\loc
\_no
\co

.
\g B3p
\ge for them (plural)
\type s
\a nagi
\a nag  / _ [V]
\u nagi
\mp MC1
\o 60
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g B3s
\ge for him/her/it
\type s
\a fu
\u fu
\mp MC2
\o 60
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g bad
\ge bad
\type r
\a inyaba
\u inyaba
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g be
\ge be
```

```
\type r
\a inyi
\a iny  / _ [V]
\u inyi
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.bad
\ge become.bad
\type r
\a inyaba   +/ ~_ TO
\u inyaba_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.crooked
\ge become.crooked
\type r
\a kobibi   +/ ~_ TO
\u kobibi_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.good
\ge become.good
\type r
\a hilou   +/ ~_ TO
\a hiloo   +/ ~_ TO
\u hilou_2
\mp MC2
\o
\c VR
\e
```

```
\mcc
\f
\loc
\_no
\co

.
\g become.heavy
\ge become.heavy
\type r
\a ou   +/ ~_ TO
\u ou_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.large
\ge become.large
\type r
\a anyakaro   +/ ~_ TO
\u anyakaro_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.many
\ge become.many
\type r
\a baingaro   +/ ~_ TO
\u baingaro_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.nothing
\ge become.nothing
```

```
\type r
\a uuwa   +/ ~_ TO
\u uuwa_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.red
\ge become.red
\type r
\a giiri   +/ ~_ TO
\u giiri_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.short
\ge become.short
\type r
\a tutu   +/ ~_ TO
\u tutu_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.small
\ge become.small
\type r
\a menekele   +/ ~_ TO
\a melekene   +/ ~_ TO
\u menekele_2
\mp MC2
\o
\c VR
\e
\mcc
```

```
\f
\loc
\_no
\co

.
\g become.straight
\ge become.straight
\type r
\a tigini   +/ ~_ TO
\u tigini_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.true
\ge become.true
\type r
\a ngalenga   +/ ~_ TO
\u ngalenga_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g become.wild
\ge become.wild
\type r
\a kui   +/ ~_ TO
\u kui_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g bound
\ge bound
\type r
```

```
\a dage
\a dag  / _ [V]
\u dage
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g break.in.two
\ge break.in.two
\type r
\a kou
\a ko
\u kou
\mp MC2 RC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g causative
\ge causative
\type s
\a mu
\u mu
\mp
\o 10
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g certainty
\ge certainty
\type s
\a nga
\u nga
\mp
\o 110
\c VSjS/VSjS
\e
\mcc
```

```
\f
\loc
\_no
\co

.
\g chase
\ge chase
\type r
\a soo
\u soo
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g cloth
\ge cloth
\type r
\a ogola
\u ogola
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g come.down
\ge come.down
\type r
\a me
\a m  / _ [V]
\u me
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g come.up
\ge come.up
```

```
\type r
\a mai
\a may  / _ [V]
\u mai
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g completely
\ge completely
\type s
\a boro
\u boro
\mp
\o
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g connect/join
\ge connect/join
\type r
\a bagumu
\u bagu-mu
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g consume
\ge consume
\type r
\a nyi
\a ny  / _ [V]
\u nyi
\mp MC1
\o
\c VR
\e
```

```
\mcc
\f
\loc
\_no
\co

.
\g Contra
\ge Contrafactual
\type s
\a ge   +/ [Contra.Subj] _
\u ge
\mp
\o 100
\c Contra/Contra
\e
\mcc
\f
\loc
\_no
\co

.
\g Contra.S1d
\ge Contractual first dual subject
\type s
\a edere  / [C] _  +/ _ Contra
\a wedere  / [V] _  +/ _ Contra
\u edere
\mp
\o 80
\c VR/Contra SMV/Contra
\e
\mcc
\f
\loc
\_no
\co

.
\g Contra.S1p
\ge Contractual first plural subject
\type s
\a ede  / [C] _  +/ _ Contra
\a wede  / [V] _  +/ _ Contra
\u ede
\mp
\o 80
\c VR/Contra SMV/Contra
\e
\mcc
\f
\loc
\_no
\co

.
```

```
\g Contra.S1s
\ge Contractual first singular subject
\type s
\a e  / [C] _  +/ _ Contra
\a we  / [V] _  +/ _ Contra
\u e
\mp
\o 80
\c VR/Contra SMV/Contra
\e
\mcc
\f
\loc
\_no
\co

.
\g Contra.S2/3s/d/p
\ge Contractual second or third singular, dual,
                       or plural subject
\type s
\a a  / [C] _  +/ _ Contra
\a wa  / [V] _  +/ _ Contra
\u a
\mp
\o 80
\c VR/Contra SMV/Contra
\e
\mcc
\f
\loc
\_no
\co

.
\g crooked
\ge crooked
\type r
\a kobibi
\u kobibi
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g cut
\ge cut
\type r
\a karu
\a ka
\u karu
```

```
\mp MC2 RC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g deflect
\ge deflect
\type r
\a ariri
\u ariri
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g desire
\ge desire
\type r
\a gau
\u gau
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g die
\ge die
\type r
\a ume
\a um  / _ [V]
\u ume
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
```

```
\co

.
\g different
\ge different
\type r
\a fere
\u fere
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g do.not
\ge do.not
\type r
\a adai
\u adai
\mp
\o
\c Adv
\e
\mcc
\f
\loc
\_no
\co

.
\g do.thus
\ge do.thus
\type r
\a ono
\u ono
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g do.work
\ge do.work
\type r
\a haruwe   +/ ~_ TO
\u haruwe_2
\mp MC2
```

```
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g dream
\ge dream
\type r
\a wewe
\u wewe
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g encircle
\ge encircle
\type r
\a kati
\u kati
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g face
\ge face
\type r
\a hogo
\u hogo
\mp
\o
\c PNR
\e
\mcc
\f
\loc
\_no
\co

```

```
.
\g fear
\ge fear
\type r
\a umuge
\a umug  / _ [V]
\u umuge
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g fight
\ge fight
\type r
\a wara
\a war  / _ [V]
\u wara
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g fill
\ge fill
\type r
\a bolo
\u bolo
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g finish
\ge finish
\type r
\a usu
\u usu
\mp MC2
```

```
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g fly
\ge fly
\type r
\a fililai
\u fililai
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g follow
\ge follow
\type r
\a oojo
\u oojo
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g gather
\ge gather
\type r
\a elege
\a eleg  / _ [V]
\u elege
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co
```

```

.
\g get.O1d
\ge get us (dual)
\type r
\a haire
\a hair  / _ [V]
\u haire
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g get.O1p
\ge get us (plural)
\type r
\a haige
\a haig  / _ [V]
\u haige
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g get.O1s
\ge get me
\type r
\a haife
\a haif  / _ [V]
\u haife
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g get.O2d
\ge get you (dual)
\type r
\a hatire
```

```
(7798)      \a hatir  / _ [V]
(7799)      \u hatire
(7800)      \mp MC1
(7801)      \o
(7802)      \c VR
(7803)      \e
(7804)      \mcc
(7805)      \f
(7806)      \loc
(7807)      \_no
(7808)      \co
(7809)
(7810)      .
(7811)      \g get.O2p
(7812)      \ge get you (plural)
(7813)      \type r
(7814)      \a hatige
(7815)      \a hatig  / _ [V]
(7816)      \u hatige
(7817)      \mp MC1
(7818)      \o
(7819)      \c VR
(7820)      \e
(7821)      \mcc
(7822)      \f
(7823)      \loc
(7824)      \_no
(7825)      \co
(7826)
(7827)      .
(7828)      \g get.O2s
(7829)      \ge get you (singular)
(7830)      \type r
(7831)      \a haine
(7832)      \a hain  / _ [V]
(7833)      \u haine
(7834)      \mp MC1
(7835)      \o
(7836)      \c VR
(7837)      \e
(7838)      \mcc
(7839)      \f
(7840)      \loc
(7841)      \_no
(7842)      \co
(7843)
(7844)      .
(7845)      \g get.O3d
(7846)      \ge get them (dual)
(7847)      \type r
(7848)      \a hanire
(7849)      \a hanir  / _ [V]
(7850)      \u hanire
(7851)      \mp MC1
(7852)      \o
(7853)      \c VR
(7854)      \e
```

```
\mcc
\f
\loc
\_no
\co

.
\g get.O3p
\ge get them (plural)
\type r
\a hanige
\a hanig  / _ [V]
\u hanige
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g get.O3s
\ge get him/her/it
\type r
\a hau
\u hau
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g get.up
\ge get.up
\type r
\a jai
\a jay  / _ [V]
\u jai
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
```

```
\g get/take
\ge get/take
\type r
\a te
\a t  / _ [V]
\u te
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g give.O1d
\ge give us (dual)
\type r
\a hiri
\a hir  / _ [V]
\u hiri
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g give.O1p
\ge give us (plural)
\type r
\a higi
\a hig  / _ [V]
\u higi
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g give.O1s
\ge give me
\type r
\a hii
\a hiy  / _ [V]
\u hii
```

```
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g give.O2d
\ge give you (dual)
\type r
\a tiri
\a tir  / _ [V]
\u tiri
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g give.O2p
\ge give you (plural)
\type r
\a tigi
\a tig  / _ [V]
\u tigi
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g give.O2s
\ge give you (singular)
\type r
\a hini
\a hin  / _ [V]
\u hini
\mp MC1
\o
\c VR
\e
\mcc
\f
```

```
\loc
\_no
\co

.
\g give.O3d
\ge give them (dual)
\type r
\a niri
\a nir  / _ [V]
\u niri
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g give.O3p
\ge give them (plural)
\type r
\a nigi
\a nig  / _ [V]
\u nigi
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g give.O3s
\ge give him/her/it
\type r
\a tu
\u tu
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g go
\ge go
```

```
\type r
\a goi
\a goy  / _ [V]
\u goi
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g go.down
\ge go.down
\type r
\a mini
\a min  / _ [V]
\u mini
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g good
\ge good
\type r
\a hilou
\a hiloo
\u hilou
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g Hab
\ge Habitual
\type s
\a g
\u g
\mp
\o 70
\c VR/SFV VR/Nom
```

```
\e
\mcc
\f
\loc
\_no
\co

.
\g hand
\ge hand
\type r
\a owo
\u owo
\mp
\o
\c PNR
\e
\mcc
\f
\loc
\_no
\co

.
\g he/she/it
\ge he/she/it
\type r
\a no
\u no
\mp
\o
\c PerPro
\e
\mcc
\f
\loc
\_no
\co

.
\g hear/understand/obey
\ge hear/understand/obey
\type r
\a isi
\a is  / _ [V]
\u isi
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
```

```
\g heavy
\ge heavy
\type r
\a ou
\u ou
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g help
\ge help
\type r
\a hoyo
\u hoyo
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g himself/herself/itself
\ge himself/herself/itself
\type r
\a nogo
\u nogo
\mp
\o
\c RflPro
\e
\mcc
\f
\loc
\_no
\co

.
\g his/her/its
\ge his/her/its
\type r
\a nomo
\u nomo
\mp
\o
\c PosPro
\e
```

```
\mcc
\f
\loc
\_no
\co

.
\g hit.O1d
\ge hit us (dual)
\type r
\a harire
\a harir  / _ [V]
\u harire
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g hit.O1p
\ge hit us (plural)
\type r
\a harige
\a harig  / _ [V]
\u harige
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g hit.O1s
\ge hit me
\type r
\a yari
\a yar  / _ [V]
\u yari
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

```

```
.
\g hit.O2d
\ge hit you (dual)
\type r
\a tarire
\a tarir  / _ [V]
\u tarire
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g hit.O2p
\ge hit you (plural)
\type r
\a tarige
\a tarig  / _ [V]
\u tarige
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g hit.O2s
\ge hit you (singular)
\type r
\a nari
\a nar  / _ [V]
\u nari
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g hit.O3d
\ge hit them (dual)
\type r
\a narire
\a narir  / _ [V]
```

```
\u narire
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g hit.O3p
\ge hit them (plural)
\type r
\a narige
\a narig  / _ [V]
\u narige
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g hit.O3s
\ge hit him/her/it
\type r
\a wari
\a war  / _ [V]
\u wari
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g I
\ge I
\type r
\a ji
\u ji
\mp
\o
\c PerPro
\e
\mcc
\f
```

```
\loc
\_no
\co

.
\g idle
\ge idle
\type r
\a ewe
\u ewe
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.S1d
\ge Indicative first dual subject
\type s
\a 0  / o _ #  +/ Ind.Tf _ ~SR
\a dere  / e _  +/ Ind.Tf _ SR
\a re  / e _  +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\a yare   +/ [Ind.Tip.Tp] _
\a yare  / o _  +/ Ind.Tf _ ~SR
\u yare
\mp
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.S1p
\ge Indicative first plural subject
\type s
\a 0  / o _ #  +/ Ind.Tf _ ~SR
\a de  / e _  +/ Ind.Tf _ SR
\a ya   +/ [Ind.Tip.Tp] _
\a ya  / o _  +/ Ind.Tf _ ~SR
\a ye  / e _  +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\u ya
\mp
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV
\e
\mcc
\f
\loc
```

```
\_no
\co

.
\g Ind.S1s
\ge Indicative first singular subject
\type s
\a he  / e _  +/ Ind.Tf _ SR
\a n  / o _ #  +/ Ind.Tf _ ~SR
\a na   +/ Ind.Tip _
\a ne   +/ [Ind.Tip.Tp] _ SR
\a ne  / e _  +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\a ni   +/ Ind.Tp _ ~SR +/ [Ind.Tip.Tp] _ SR
\a ni  / o _  +/ Ind.Tf _ ~SR
\u ni
\mp
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.S2d
\ge Indicative second dual subject
\type s
\a dere  / a _  +/ Ind.Tf _ SR
\a re  / e _  +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\a yare   +/ [Ind.Tip.Tp] _
\u yare
\mp
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.S2p
\ge Indicative second plural subject
\type s
\a de  / a _  +/ Ind.Tf _ SR
\a ya   +/ [Ind.Tip.Tp] _
\a ye  / e _  +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\u ya
\mp
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV
\e
\mcc
\f
```

```
\loc
\_no
\co

.
\g Ind.S2s
\ge Indicative second singular subject
\type s
\a he  / a _  +/ Ind.Tf _ SR
\a na   +/ Ind.Tip _
\a ne   +/ Ind.Tp _ SR
\a ne  / e _  +/ [Ind.Tip.Tp] _ SR +/ Ind.Trp _
\a ni   +/ Ind.Tp _ ~SR +/ [Ind.Tip.Tp] _ SR
\u ni
\mp
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.S3d
\ge Indicative third dual subject
\type s
\a d  / o _ #  +/ Ind.Tf _ ~SR
\a dere   +/ Ind.Tp _ ~SR
\a dere  / o _  +/ Ind.Tf _ ~SR
\a re  / a _  +/ Ind.Tip _ +/ Ind.Tp _ SR +/ Ind.Trp _
\a ro  / o _  +/ Ind.Tf _ SR
\u dere
\mp
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.S3p
\ge Indicative third plural subject
\type s
\a d  / o _ #  +/ Ind.Tf _ ~SR
\a de   +/ Ind.Tp _ ~SR
\a de  / o _  +/ Ind.Tf _
\a go  / o _  +/ Ind.Tf _ SR
\a ne  / a _  +/ Ind.Tip _ +/ Ind.Tp _ SR +/ Ind.Trp _
\u de
\mp
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV
```

```
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.S3s
\ge Indicative third singular subject
\type s
\a 0  / a _  +/ Ind.Tip _ +/ Ind.Tp _ SR
\a 0  / o _  +/ Ind.Tf _ SR
\a i  / a _  +/ Ind.Trp _
\a n  / o _ #  +/ Ind.Tf _ ~SR
\a na   +/ Ind.Tp _ ~SR
\a ta   +/ Ind.Tip _ ~SR
\a na  / o _  +/ Ind.Tf _ ~SR
\u na
\mp
\o 90
\c SFV/SFV SFV/SFVW SMV/SMV
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.Tf
\ge Indicative future tense
\type s
\a e  / [C] _  +/ _ [Ind.S1]
\a we  / [V] _  +/ _ [Ind.S1]
\a a  / [C] _  +/ _ [Ind.S2]
\a wa  / [V] _  +/ _ [Ind.S2]
\a o  / [C] _  +/ _ [Ind.S1.S3]
\a wo  / [V] _  +/ _ [Ind.S1.S3]
\u o
\mp
\o 80
\c SMV/SMV VR/SMV VR/SFV SFV/SFV
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.Tip
\ge Indicative immediate past tense
\type s
\a e  / [C] _  +/ _ [Ind.S1.S2]
\a we  / [V] _  +/ _ [Ind.S1.S2]
\a a  / [C] _  +/ _ [Ind.S3]
```

```
\a wa  / [V] _  +/ _ [Ind.S3]
\a 0   +/ _ [Ind.S1.S2] +/ _ Ind.S3s
\u a
\mp
\o 80
\c SMV/SMV VR/SMV VR/SFV SFV/SFV
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.Tp
\ge Indicative present tense
\type s
\a 0   +/ _ [Ind.S1.S2.S3]
\a e  / [C] _  +/ _ [Ind.S1.S2]
\a we  / [V] _  +/ _ [Ind.S1.S2]
\a a  / [C] _  +/ _ [Ind.S3]
\a wa  / [V] _  +/ _ [Ind.S3]
\u 0
\mp
\o 80
\c SMV/SMV VR/SMV VR/SFV SFV/SFV
\e
\mcc
\f
\loc
\_no
\co

.
\g Ind.Trp
\ge Indicative remote past tense
\type s
\a e  / [C] _  +/ _ [Ind.S1.S2]
\a we  / [V] _  +/ _ [Ind.S1.S2]
\a a  / [C] _  +/ _ [Ind.S3]
\a wa  / [V] _  +/ _ [Ind.S3]
\u e
\mp
\o 80
\c SMV/SMV VR/SMV VR/SFV SFV/SFV
\e
\mcc
\f
\loc
\_no
\co

.
\g Inf
\ge Infinitive
\type s
\a e  / [C] _
```

```
\a we  / [V] _
\u e
\mp
\o 80
\c VR/Inf
\e
\mcc
\f
\loc
\_no
\co

.
\g Inf.Sd
\ge Infinitive subject
\type s
\a ru
\u ru
\mp
\o 90
\c Inf/Inf
\e
\mcc
\f
\loc
\_no
\co

.
\g Inf.Sp
\ge Infinitive plural subject
\type s
\a gu
\u gu
\mp
\o 90
\c Inf/Inf
\e
\mcc
\f
\loc
\_no
\co

.
\g Inf.Ss
\ge Infinitive singular subject
\type s
\a i
\u i
\mp
\o 90
\c Inf/Inf
\e
\mcc
\f
\loc
```

```
(8767)      \_no
(8768)      \co
(8769)
(8770)      .
(8771)      \g inside
(8772)      \ge inside
(8773)      \type r
(8774)      \a huwanya
(8775)      \u huwanya
(8776)      \mp
(8777)      \o
(8778)      \c PNR
(8779)      \e
(8780)      \mcc
(8781)      \f
(8782)      \loc
(8783)      \_no
(8784)      \co
(8785)
(8786)      .
(8787)      \g judge
(8788)      \ge judge
(8789)      \type r
(8790)      \a haawe
(8791)      \u haawe
(8792)      \mp MC2
(8793)      \o
(8794)      \c VR
(8795)      \e
(8796)      \mcc
(8797)      \f
(8798)      \loc
(8799)      \_no
(8800)      \co
(8801)
(8802)      .
(8803)      \g just/only
(8804)      \ge just/only
(8805)      \type s
(8806)      \a nga
(8807)      \u nga
(8808)      \mp
(8809)      \o
(8810)      \c NW/NW RflPro/RflPro PN/PN Adj/Adj PosPro/PosPro
   PerPro/PerPro Prep/Prep Deictic/Deictic VR/VR
(8811)      \e
(8812)      \mcc
(8813)      \f
(8814)      \loc
(8815)      \_no
(8816)      \co
(8817)
(8818)      .
(8819)      \g knife
(8820)      \ge knife
(8821)      \type r
(8822)      \a kesebu
```

```
\u kesebu
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g large
\ge large
\type r
\a anyakaro
\u anyakaro
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g leg/foot
\ge leg/foot
\type r
\a afe
\u afe
\mp
\o
\c PNR
\e
\mcc
\f
\loc
\_no
\co

.
\g leg/foot.3sP
\ge his/her/its leg/foot
\type r
\a afo
\u afo
\mp
\o
\c PNR
\e
\mcc
\f
\loc
\_no
```

```
\co

.
\g lie
\ge lie
\type r
\a yaawa
\u yaawa
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g light
\ge light
\type r
\a lala
\u lala_1
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g like.that
\ge like.that
\type r
\a onou
\u onou
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g like1
\ge like1
\type s
\a fe
\u fe
\mp
```

```
\o 100
\c NW/NW Deictic/Deictic Adj/Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g like2
\ge like2
\type s
\a neu   +/ this _
\u neu
\mp
\o 100
\c Deictic/Deictic
\e
\mcc
\f
\loc
\_no
\co

.
\g look.for
\ge look.for
\type r
\a kuru
\u kuru
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g man
\ge man
\type r
\a fai
\u fai
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

```

```
.
\g many
\ge many
\type r
\a baingaro
\u baingaro
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g my
\ge my
\type r
\a yame
\u yame
\mp
\o
\c PosPro
\e
\mcc
\f
\loc
\_no
\co

.
\g myself
\ge myself
\type r
\a yage
\u yage
\mp
\o
\c RflPro
\e
\mcc
\f
\loc
\_no
\co

.
\g N.not
\ge not
\type r
\a uuwa
\u uuwa_1
\mp
\o
\c NomNeg
```

```
\e
\mcc
\f
\loc
\_no
\co

.
\g Nom
\ge Nominalizer
\type s
\a ou  / [C] _
\a wou  / [V] _
\u ou
\mp
\o 110
\c VR/Nom Nom/Nom
\e
\mcc
\f
\loc
\_no
\co

.
\g nose
\ge nose
\type r
\a mutu
\u mutu
\mp
\o
\c PNR
\e
\mcc
\f
\loc
\_no
\co

.
\g O1d
\ge to us (dual)
\type s
\a hiri
\a hir  / _ [V]
\u hiri
\mp MC1
\o 30
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

```

```
.
\g O1p
\ge to us (plural)
\type s
\a higi
\a hig  / _ [V]
\u higi
\mp MC1
\o 30
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g O1s
\ge to me
\type s
\a hi
\a hiy  / _ [V]
\u hii
\mp MC2
\o 30
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g O2d
\ge to you (dual)
\type s
\a tiri
\a tir  / _ [V]
\u tiri
\mp MC1
\o 30
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g O2p
\ge to you (plural)
\type s
\a tigi
\a tig  / _ [V]
```

```
\u tigi
\mp MC1
\o 30
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g O2s
\ge to you (singular)
\type s
\a hini
\a hin  / _ [V]
\u hini
\mp MC1
\o 30
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g O3d
\ge to them (dual)
\type s
\a niri
\a nir  / _ [V]
\u niri
\mp MC1
\o 30
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g O3p
\ge to them (plural)
\type s
\a nigi
\a nig  / _ [V]
\u nigi
\mp MC1
\o 30
\c VR/VR
\e
\mcc
```

```
\f
\loc
\_no
\co

.
\g O3s
\ge to him/her/it
\type s
\a tu
\u tu
\mp MC2
\o 30
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g ocean
\ge ocean
\type r
\a hiiri
\u hiiri
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g one/a
\ge one/a
\type r
\a ete
\u ete
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g our.dual
\ge our.dual
\type r
```

```
\a heire
\u heire
\mp
\o
\c PosPro
\e
\mcc
\f
\loc
\_no
\co

.
\g our.plural
\ge our.plural
\type r
\a ebere
\u ebere
\mp
\o
\c PosPro
\e
\mcc
\f
\loc
\_no
\co

.
\g ourselves.dual
\ge ourselves.dual
\type r
\a hare
\u hare
\mp
\o
\c RflPro
\e
\mcc
\f
\loc
\_no
\co

.
\g ourselves.plural
\ge ourselves.plural
\type r
\a hage
\u hage
\mp
\o
\c RflPro
\e
\mcc
\f
\loc
```

```
\_no
\co

.
\g p1
\ge plural
\type s
\a gana
\u gana
\mp
\o 20
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g P1d
\ge first dual possessive
\type s
\a re
\u re
\mp
\o 10
\c PNR/PN
\e
\mcc
\f
\loc
\_no
\co

.
\g P1p
\ge first plural possessive
\type s
\a ge
\u ge
\mp
\o 10
\c PNR/PN
\e
\mcc
\f
\loc
\_no
\co

.
\g P1s
\ge first singular possessive
\type s
\a fe
\u fe
```

```
(9393)      \mp
(9394)      \o 10
(9395)      \c PNR/PN
(9396)      \e
(9397)      \mcc
(9398)      \f
(9399)      \loc
(9400)      \_no
(9401)      \co
(9402)
(9403)      .
(9404)      \g p2
(9405)      \ge plural
(9406)      \type s
(9407)      \a ru
(9408)      \u ru
(9409)      \mp
(9410)      \o 20
(9411)      \c VR/VR
(9412)      \e
(9413)      \mcc
(9414)      \f
(9415)      \loc
(9416)      \_no
(9417)      \co
(9418)
(9419)      .
(9420)      \g P2d
(9421)      \ge second dual possessive
(9422)      \type s
(9423)      \a tere
(9424)      \u tere
(9425)      \mp
(9426)      \o 10
(9427)      \c PNR/PN
(9428)      \e
(9429)      \mcc
(9430)      \f
(9431)      \loc
(9432)      \_no
(9433)      \co
(9434)
(9435)      .
(9436)      \g P2p
(9437)      \ge second plural possessive
(9438)      \type s
(9439)      \a tege
(9440)      \u tege
(9441)      \mp
(9442)      \o 10
(9443)      \c PNR/PN
(9444)      \e
(9445)      \mcc
(9446)      \f
(9447)      \loc
(9448)      \_no
(9449)      \co
```

```

.
\g P2s
\ge second singular possessive
\type s
\a ne
\a te
\u ne
\mp
\o 10
\c PNR/PN
\e
\mcc
\f
\loc
\_no
\co

.
\g P3d
\ge third dual possessive
\type s
\a nere
\u nere
\mp
\o 10
\c PNR/PN
\e
\mcc
\f
\loc
\_no
\co

.
\g P3p
\ge third plural possessive
\type s
\a nege
\u nege
\mp
\o 10
\c PNR/PN
\e
\mcc
\f
\loc
\_no
\co

.
\g P3s
\ge third singular possessive
\type s
\a 0
\a u
\u 0
```

```
\mp
\o 10
\c PNR/PN
\e
\mcc
\f
\loc
\_no
\co

.
\g perf
\ge perfective
\type s
\a ha
\u ha
\mp
\o 100
\c SFV/SFVW
\e
\mcc
\f
\loc
\_no
\co

.
\g pertaining.to
\ge pertaining.to
\type s
\a ha
\u ha
\mp
\o
\c Adj/Adj NW/NW Deictic/Deictic PosPro/PosPro
\e
\mcc
\f
\loc
\_no
\co

.
\g pig
\ge pig
\type r
\a buuwa
\u buuwa
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co
```

```

.
\g place
\ge place
\type r
\a ha
\a haumu
\u ha
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g plant
\ge plant
\type r
\a tau
\a ta
\u tau
\mp MC2 RC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g poke
\ge poke
\type r
\a tu  / _ [O]
\a tum  / _ [B]
\a tun  / _ [A]
\a tung  / _ [X]
\u tuN
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g pour.out
\ge pour.out
```

```
\type r
\a feu
\u feu
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g Prediction
\ge Prediction
\type s
\a ga
\u ga
\mp
\o 100
\c VSjS/VSjS
\e
\mcc
\f
\loc
\_no
\co

.
\g prepare
\ge prepare
\type r
\a haiya
\u haiya
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g press
\ge press
\type r
\a ilei
\u ilei
\mp MC2
\o
\c VR
\e
\mcc
\f
```

```
\loc
\_no
\co

.
\g put
\ge put
\type r
\a tafa
\a taf  / _ [V]
\u tafa
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g quickly
\ge quickly
\type r
\a fasa
\u fasa
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g quickly
\ge quickly
\type s
\a ya
\u ya
\mp
\o
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g red
\ge red
\type r
```

```
\a giiri
\u giiri
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g remove
\ge remove
\type r
\a uulu
\u uulu
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g run.away
\ge run.away
\type r
\a faga
\u faga
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g see.O1d
\ge see us (dual)
\type r
\a herire
\a herir  / _ [V]
\u herire
\mp MC1
\o
\c VR
\e
\mcc
\f
```

```
\loc
\_no
\co

.
\g see.O1p
\ge see us (plural)
\type r
\a herige
\a herig  / _ [V]
\u herige
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g see.O1s
\ge see me
\type r
\a yeriye
\a yeriy  / _ [V]
\u yeriye
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g see.O2d
\ge see you (dual)
\type r
\a terire
\a terir  / _ [V]
\u terire
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g see.O2p
```

```
\ge see you (plural)
\type r
\a terige
\a terig  / _ [V]
\u terige
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g see.O2s
\ge see you (singular)
\type r
\a neriye
\a neriy  / _ [V]
\u neriye
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g see.O3d
\ge see them (dual)
\type r
\a nerire
\a nerir  / _ [V]
\u nerire
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g see.O3p
\ge see them (plural)
\type r
\a nerige
\a nerig  / _ [V]
\u nerige
\mp MC1
```

```
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g see.O3s
\ge see him/her/it
\type r
\a ure
\a ur  / _ [V]
\u ure
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g send
\ge send
\type r
\a huuru
\u huuru
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g shine
\ge shine
\type r
\a lala
\u lala_2
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co
```

```

.
\g short
\ge short
\type r
\a tutu
\u tutu
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g shove
\ge shove
\type r
\a nuu
\u nuu
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g show
\ge show
\type r
\a abiti
\u abiti
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g sibling.same.sex.his/her/its
\ge sibling.same.sex.his/her/its
\type r
\a bemu
\u bai_2
\mp
\o
```

```
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g sibling.same.sex.possessive
\ge sibling.same.sex.possessive
\type r
\a bai   +/ ~_ P3s
\u bai_2
\mp
\o
\c PNR
\e
\mcc
\f
\loc
\_no
\co

.
\g sibling.same.sex.referential
\ge sibling.same.sex.referential
\type r
\a babo
\u bai_2
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g sit
\ge sit
\type r
\a yafa
\a yaf  / _ [V]
\u yafa
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

```

```
.
\g Sj.S1/3
\ge subjunctive first or third subject
\type s
\a o  / [C] _
\a wo  / [V] _
\u o
\mp
\o 80
\c VR/VSj
\e
\mcc
\f
\loc
\_no
\co

.
\g Sj.S2
\ge subjunctive second subject
\type s
\a a  / [C] _
\a wa  / [V] _
\u a
\mp
\o 80
\c VR/VSj
\e
\mcc
\f
\loc
\_no
\co

.
\g Sj.Sd
\ge subjunctive dual subject
\type s
\a ru
\u ru
\mp
\o 90
\c VSj/VSjS
\e
\mcc
\f
\loc
\_no
\co

.
\g Sj.Sp
\ge subjunctive plural subject
\type s
\a gu
\u gu
\mp
```

```
\o 90
\c VSj/VSjS
\e
\mcc
\f
\loc
\_no
\co

.
\g Sj.Ss
\ge subjunctive singular subject
\type s
\a u
\u u
\mp
\o 90
\c VSj/VSjS
\e
\mcc
\f
\loc
\_no
\co

.
\g sleep.pl
\ge sleep.pl
\type r
\a jiri
\u jiri
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g small
\ge small
\type r
\a menekele
\a melekene
\u menekele
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co
```

```

.
\g smooth
\ge smooth
\type r
\a fii
\u fii
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g some
\ge some
\type r
\a ila
\u ila
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g speak
\ge speak
\type r
\a le
\u le
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g SR
\ge switch referent
\type s
\a nga
\u nga
\mp
\o 110
```

```
\c SMV/SMVW Contra/SMVW
\e
\mcc
\f
\loc
\_no
\co

.
\g SR.TO
\ge switch referent with temporal overlap
\type s
\a g   +/ _ [Ind.T] +/ _ [Contra.Subj]
\u g
\mp
\o 70
\c VR/SMV
\e
\mcc
\f
\loc
\_no
\co

.
\g SR.TS
\ge switch referent with temporal succession
\type s
\a 0   +/ _ [Ind.T] +/ _ [Contra.Subj]
\u 0
\mp
\o 70
\c VR/SMV
\e
\mcc
\f
\loc
\_no
\co

.
\g stand
\ge stand
\type r
\a oto
\u oto
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
```

```
\g stick.to
\ge stick.to
\type r
\a fada
\u fada
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g stone
\ge stone
\type r
\a megebu
\u megebu
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g straight
\ge straight
\type r
\a tigini
\u tigini
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g talk/word
\ge talk/word
\type r
\a beele
\u beele
\mp
\o
\c NW
\e
```

```
\mcc
\f
\loc
\_no
\co

.
\g teach
\ge teach
\type r
\a agotete
\u agotete
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g tear
\ge tear
\type r
\a korai
\u korai
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g tell
\ge tell
\type r
\a hai
\u hai
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g that
\ge that
```

```
\type r
\a wa
\u wa
\mp
\o
\c Deictic
\e
\mcc
\f
\loc
\_no
\co

.
\g their.dual
\ge their.dual
\type r
\a neire
\u neire
\mp
\o
\c PosPro
\e
\mcc
\f
\loc
\_no
\co

.
\g their.plural
\ge their.plural
\type r
\a nebere
\u nebere
\mp
\o
\c PosPro
\e
\mcc
\f
\loc
\_no
\co

.
\g theirselves.dual
\ge theirselves.dual
\type r
\a nare
\u nare
\mp
\o
\c RflPro
\e
\mcc
\f
```

```
\loc
\_no
\co

.
\g theirselves.plural
\ge theirselves.plural
\type r
\a nage
\u nage
\mp
\o
\c RflPro
\e
\mcc
\f
\loc
\_no
\co

.
\g they
\ge they
\type r
\a nere
\u nere
\mp
\o
\c PerPro
\e
\mcc
\f
\loc
\_no
\co

.
\g thigh
\ge thigh
\type r
\a gaigu
\u gaigu
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g thing
\ge thing
\type r
\a dada
```

```
\u dada
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g this
\ge this
\type r
\a be
\u be
\mp
\o
\c Deictic
\e
\mcc
\f
\loc
\_no
\co

.
\g thought
\ge thought
\type r
\a nina
\u nina
\mp
\o
\c PNR
\e
\mcc
\f
\loc
\_no
\co

.
\g throw
\ge throw
\type r
\a aya
\u aya
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
```

```
\co

.
\g TO
\ge temporal overlap
\type s
\a 0
\u 0
\mp
\o 70
\c VR/SMVW
\e
\mcc
\f
\loc
\_no
\co

.
\g true
\ge true
\type r
\a ngalenga
\u ngalenga
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g TS
\ge temporal succession
\type s
\a du
\u du
\mp
\o 70
\c VR/SMVW
\e
\mcc
\f
\loc
\_no
\co

.
\g turn.back.on
\ge turn.back.on
\type r
\a moge
\u moge
\mp MC2
```

```
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g untie
\ge untie
\type r
\a fuutu
\u fuutu
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g V.not
\ge not
\type r
\a ada
\u ada
\mp
\o
\c VNeg
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.S1d
\ge verbal auxiliary first dual subject
\type s
\a yare
\u yare
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

```

```
.
\g vAux.S1p
\ge verbal auxiliary first plural subject
\type s
\a ya
\u ya
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.S1s
\ge verbal auxiliary first singular subject
\type s
\a ni
\u ni
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.S3d
\ge verbal auxiliary third dual subject
\type s
\a dere
\u dere
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.S3p
\ge verbal auxiliary third plural subject
\type s
\a de
\u de
\mp
\o 90
\c vAuxR/vAuxW
```

```
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.S3s
\ge verbal auxiliary third singular subject
\type s
\a na
\u na
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.Tf.S2d
\ge verbal auxiliary second dual subject
\type s
\a nadere
\u nadere
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.Tf.S2p
\ge verbal auxiliary second plural subject
\type s
\a nade
\u nade
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.Tf.S2s
```

```
\ge verbal auxiliary second singular subject, future tense
\type s
\a nahe
\u nahe
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.Tnf.S2d
\ge verbal auxiliary second dual subject, non-future tense
\type s
\a yare
\u yare
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.Tnf.S2p
\ge verbal auxiliary second plural subject, non-future tense
\type s
\a ya
\u ya
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
\f
\loc
\_no
\co

.
\g vAux.Tnf.S2s
\ge verbal auxiliary second singular subject, non-future
\type s
\a ne
\u ne
\mp
\o 90
\c vAuxR/vAuxW
\e
\mcc
```

```
\f
\loc
\_no
\co

.
\g vAuxR
\ge verbal auxiliary root
\type r
\a bo
\u bo
\mp
\o
\c vAuxR
\e
\mcc
\f
\loc
\_no
\co

.
\g very
\ge very
\type r
\a bai
\u bai
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g village
\ge village
\type r
\a matane
\u matane
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g walk
\ge walk
\type r
```

```
\a golo
\u golo
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g watch
\ge watch
\type r
\a fele
\u fele
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g watch.over
\ge watch.over
\type r
\a oofa
\u oofa
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g we
\ge we
\type r
\a ere
\u ere
\mp
\o
\c PerPro
\e
\mcc
\f
\loc
```

```
\_no
\co

.
\g well
\ge well
\type s
\a de
\u de
\mp
\o
\c VR/VR
\e
\mcc
\f
\loc
\_no
\co

.
\g what
\ge what
\type r
\a taate
\u taate
\mp
\o
\c IP
\e
\mcc
\f
\loc
\_no
\co

.
\g wild
\ge wild
\type r
\a kui
\u kui
\mp
\o
\c Adj
\e
\mcc
\f
\loc
\_no
\co

.
\g with
\ge with
\type r
\a bagu   +/ ~_ causative
\u bagu
```

```
\mp
\o
\c Prep
\e
\mcc
\f
\loc
\_no
\co

.
\g with.dual
\ge with.dual
\type r
\a ngare
\u ngare
\mp
\o
\c Prep
\e
\mcc
\f
\loc
\_no
\co

.
\g without
\ge without
\type r
\a uwau
\u uwau
\mp
\o
\c Prep
\e
\mcc
\f
\loc
\_no
\co

.
\g woman
\ge woman
\type r
\a aita
\u aita
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co
```

```

.
\g work
\ge work
\type r
\a haruwe
\u haruwe
\mp
\o
\c NW
\e
\mcc
\f
\loc
\_no
\co

.
\g write
\ge write
\type r
\a yerege
\a yereg  / _ [V]
\u yerege
\mp MC1
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g yell.to/shout
\ge yell.to/shout
\type r
\a yaure
\u yaure
\mp MC2
\o
\c VR
\e
\mcc
\f
\loc
\_no
\co

.
\g you.plural
\ge you.plural
\type r
\a tere
\u tere
\mp
```

```
\o
\c PerPro
\e
\mcc
\f
\loc
\_no
\co

.
\g you.sing
\ge you.sing
\type r
\a ne
\u ne
\mp
\o
\c PerPro
\e
\mcc
\f
\loc
\_no
\co

.
\g your.dual
\ge your.dual
\type r
\a teire
\u teire
\mp
\o
\c PosPro
\e
\mcc
\f
\loc
\_no
\co

.
\g your.plural
\ge your.plural
\type r
\a tebere
\u tebere
\mp
\o
\c PosPro
\e
\mcc
\f
\loc
\_no
\co

```

```
.
\g your.sing
\ge your.sing
\type r
\a name
\u name
\mp
\o
\c PosPro
\e
\mcc
\f
\loc
\_no
\co

.
\g yourselves.dual
\ge yourselves.dual
\type r
\a tare
\u tare
\mp
\o
\c RflPro
\e
\mcc
\f
\loc
\_no
\co

.
\g yourselves.plural
\ge yourselves.plural
\type r
\a tage
\u tage
\mp
\o
\c RflPro
\e
\mcc
\f
\loc
\_no
\co

.
```

APPENDIX 4

THE YALÁLAG ZAPOTEC AMPLE AND DATR LEXICONS

A4.1 Overview

This appendix provides listings of the following: A subset of the Yalálag Zapotec AMPLE root database file, the Yalálag Zapotec DATR "theory" file (the lexicon), the Yalálag Zapotec DATR query file, a DOS batch file that invokes DATR, and the result of running DATR using the query file--a query trace file.

For research purposes, all Order Class 10 root entries and some Order Class 9 entries were extraced from a Yalálag Zapotec AMPLE root database file supplied to this author by Dr. H. Andrew Black. The AMPLE database file shown in this appendix lists the extracted entries.

The Yalálag Zapotec DATR theory file is the heart of the Yalálag Zapotec DATR lexicon. It is divided into the following sections: function library, interfaces, templates, categories, and Yalálag Zapotec root nodes. The theory file was developed specifically to show that there are generalizations not captured in the AMPLE root file that can be captured in DATR, thereby reducing redundancy. The theory file includes an interface between DATR and AMPLE though this was not the primary focus of the DATR work done on Yalálag Zapotec.

The Yalálag Zapotec DATR query file is the driver for the interface between DATR and AMPLE. The order of the entries in the query file mirror the order of the entries in the Yalálag Zapotec AMPLE root database file.

The DOS batch file listing contains two calls to DATR. The first call creates a token file, the second executes the query file against the token file.

The Yalálag Zapotec DATR trace file is the output of running the DOS batch file. The trace file provides the Yalálag Zapotec AMPLE lexical entries.

A4.2 The Yalálag Zapotec AMPLE Root Database File

A4.2.1 Description

A4.2.2 File Listing

Filename: yal0910.db

Note: line numbers added for reference only. Not in actual file.

```
(11266)     \r cha'a
(11267)     \_no 00129
(11268)     \u cha'
(11269)     \a cha'   /~_ #
(11270)     \a cha'a  / _ #
(11271)     \c VA
(11272)     \g estar.en.grupo
(11273)     \m 10
(11274)     \mp vi
(11275)     \mp vt
(11276)     \mp class2
(11277)     \mp takes_null_S
(11278)     \mcc estar.en.grupo / ~Pl.pfx _
(11279)
(11280)     \r chach
(11281)     \_no 00130
(11282)     \u cha:ch
(11283)     \a cha:ch
(11284)     \c VA
(11285)     \g aparecer.muchos
(11286)     \m 10
(11287)     \mp vi
(11288)     \mp class2
(11289)     \mp takes_null_S
(11290)     \co llach kwerp kello - effect of something that scares us
```

```
(11291)     \r cha'ra'll
(11292)     \_no 00138
(11293)     \u cha'la'ch
(11294)     \a cha'la'ch
(11295)     \c VA
(11296)     \g engordar
(11297)     \m 10
(11298)     \mp vi
(11299)     \mp class2
(11300)     \mp takes_null_S
(11301)     \mp takes_:_Caus

(11302)     \r chel
(11303)     \_no 00158
(11304)     \u che:l
(11305)     \a che:l
(11306)     \c VA
(11307)     \g enrollar             | take group 2 pns??
(11308)     \m 9
(11309)     \mp vi
(11310)     \mp class2
(11311)     \mp takes_null_S
(11312)     \mp takes_:_Caus
(11313)
(11314)     \r che'z
(11315)     \_no 00170
(11316)     \u che's
(11317)     \a che's    +/ [P-or-Modal]      _
(11318)     \a   e's    +/ H                 _
(11319)     \a  che's    +/ [P,H-or-Modal] ~_
(11320)     \c VA
(11321)     \g romper
(11322)     \m 9
(11323)     \mp vi
(11324)     \mp class2
(11325)     \mp takes_:_Caus
(11326)
(11327)     \r chia
(11328)     \_no 00172
(11329)     \u chia
(11330)     \a chi  / _ [V-hi+cg]   +/ _ [1s-or-3r]
(11331)     \a chia
(11332)     \c VA
(11333)     \g coser.vt
(11334)     \m 9
(11335)     \mp vt
(11336)     \mp class2
(11337)
(11338)     \r chi'o
(11339)     \_no 00195
(11340)     \u chi'o
(11341)     \a chi'  / _ o'  +/ _ 2s
(11342)     \a chi'kw / _ [V-hi+cg]   +/ _ [1s-or-3r]
(11343)     \a chi'o
(11344)     \c VA
(11345)     \g crecer
(11346)     \m 9
```

```
\mp vi
\mp class2
\mp takes_:s_Caus

\r labj
\_no 00495
\u lapj
\a lapj
\c VA
\g susmergir    | group 2 pns??
\m 10
\mp vi
\mp class2
\mp takes_null_S

\r la'll
\_no 00498
\u la'ch
\a la'ch
\c VA
\g vagar
\m 10
\mp vi
\mp class2
\mp takes_null_S

\r lap
\_no 00503
\u la:p
\a la:p
\c VA
\g pizcar
\m 10
\mp vt
\mp class2
\mp takes_null_S

\r len
\_no 00536
\u len
\a len
\c VA
\g asociar
\m 10
\mp vi
\mp class2
\mp takes_null_S
\mp takes_:_Caus

\r nit
\_no 00658
\u ni:t
\a ni:t
\c VA
\g perder.vi
\m 10
\mp vi
```

```
\mp class2
\mp takes_null_S

\r nna'lÄllej
\_no 00666
\u na':lchej
\a na':lchej
\c VA
\g sorprender
\m 10  |no Habitual
\mp vi
\mp class2
\mp takes_null_S

\r nnao
\_no 00667
\u nao
\a na  under_ao / _ o'        +/ _ 2s
\a nakw         / _ [V-hi+cg] +/ _ [1s-or-3r]
/ _ [V-hi-cg]p  | where 1s or 3r is followed by 3f or 3a
\a nao
\c VA
\g seguir
\m 10
\mp vt
\mp class2
\mp takes_null_S

\r nne
\_no 00670
\u ne
\a ni / _ a' +/ _ 1s
\a n, / _ # +/ [Modal] ... _
+/ Rep ...     _
+/ [Imperatives] ... _
+/ Pl.pfx ... _
\a ne / # ([C]) _  #
/       _ ~#
\c VA
\g decir2
\m 10
\mp vt
\mp class2
\mp takes_null_S

\r nnebia'
\_no 00672
\u nepia'
\a nepi   / _ [V-hi+cg]   +/ _ [1s-or-3r]
/ _ [V-hi-cg]p  | where 1s or 3r is followed by 3f or 3a
\a nepia'
\c VA
\g gobernar
\m 10
\mp vt
\mp class2

```

```
\r nni't
\_no 00675
\u ni':t
\a ni':t
\c VA
\g estar.regado | take group 2 pns?
\m 10
\mp vi
\mp class2
\mp takes_null_S

\r nnitrao
\_no 00676
\u ni:tlao
\a ni:tla    under_ao  / _ o'          +/ _ 2s
\a ni:tlakw            / _ [V-hi+cg]   +/ _ [1s-or-3r]
\a ni:tlao
\c VA
\g borrar.vi
\m 10
\mp vi
\mp class2
\mp takes_null_S

\r sa'a
\_no 00752
\u sa'
\a s   / _ a'  +/ _ 1s
\a se' / _ [V-hi-cg]   +/ _ [1s-or-3r]
\a sa'a / _ #
\a sa'  / ~_ #
\c VA
\g ir/salir/caminar   | caminar formerly separate record
\m 10
\mp vi
\mp class2
\mp takes_null_S
\mp laryngealizes

\r sa'ra'll
\_no 00757
\u sa'la'ch
\a sa'la'ch
\c VA
\g sospechar
\m 10
\mp vi
\mp class2
\mp takes_null_S
\co see also recordar2 wasa'ra'll

\r se
\_no 00759
\u se
\a si / _ a' +/ _ 1s
\a s, / _ # +/ [Modal] ... _
+/ Rep ...     _
+/ [Imperatives] ... _
+/ Plural.S ... _
\a s, / _ :tese| +/ [Modal] ... _
| +/ Rep ...      _
| +/ [Imperatives] ... _
| +/ Plural.S ... _
\a se / # ([C]) _  #
/          _ ~#
\c VA
\g parar.vi1
\m 10
\mp vi
\mp class2
\mp takes_null_S

\r sella
\_no 00761
\u secha
\a sech   / _ a'  +/ _ 1s
\a seche   / _ e'  +/ _ 3r
\a sech    / _ #
\a secha   / ~_ #
\a soch   / _ a'  +/ _ 1s
\a soche   / _ e'  +/ _ 3r
\a soch    / _ #
\a socha   / ~_ #
\c VA
\g parar.vi2 | Is -cha really part of root or is it an Adv? yes
\m 10
\mp vi
\mp class2
\mp takes_null_S
\noload

\r sera'll
\_no 00764
\u sela'ch
\a sela'ch
\c VA
\g desear
\m 10
\mp vt
\mp class2
\mp takes_null_S

\r sill
\_no 00773
\u sich
\a sich
\c VA
\g sonar
\m 10
\mp vi
\mp class2
\mp takes_null_S
\mp takes_:_Caus
```

A4.3 The Yalálag Zapotec DATR Theory File

A4.3.1 Description

The Yalálag Zapotec DATR theory file is divided into the following sections: function library, interfaces, templates, categories, and Yalálag Zapotec root nodes for Order Class 10 and a subset of Order Class 9.

The function library loads the standard DATR function library. The interfaces section provides a one-way interface from DATR to AMPLE. The templates section provides a series of abstract nodes to support multiple allomorph entries in a root node entry. The categories section provides abstract nodes that potentially represent grammatical and morphological categories. It is through these abstract nodes that redundancy in the Yalálag Zapotec AMPLE lexicon can be reduced for Order Class 9 and 10 root entries. The last section is Yalálag Zapotec root node section. This section contains a subset of Yalálag Zapotec roots from the Yalálag Zapotec AMPLE root database file--a subset of Order Class 9, and all of the Order Class 10 entries.

A4.3.2 File Listing

Filename: yal.dtr

Note: line numbers added for reference only. Not in actual file.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                            %
% File:        yal.dtr                                       %
% Purpose:     Partial lexicon - Yalalag                     %
%                          Zapotec - Order Class 10 roots    %
%                          and some Order Class 9 roots      %
% Yalalag Zapotec:   A Language of Mexico                    %
% Data from:   yalrt.db (from Dr. H. Andrew Black)           %
% Author:      Michael Colburn                               %
% Email:       mcolburn@sprintmail.com                       %
% Created:     November 12, 1999                             %
% Last update:                                               %
% Version:       1.0                                         %
%                                                            %
% Notes:         Limited entries -- for research and         %
%                                  illustration only         %
%                Illustrates reduction of redundancy,        %
%                capture of generalizations                  %
%                                                            %
%                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%  Function Library %%%%%%%%%%%%%%%%%%%%%
%

#load 'std.dtr' all.

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%  Interfaces  %%%%%%%%%%%%%%%%%%%%%%%%%%
%

I_AMPLE:
<amp lex morname> == <amp root>
                     <amp noload>
                     <amp underlying form>
                     "<lex form lex>"
                     <amp category>
                     <amp gloss morpheme name>
                     <amp order class>
                     <amp mor property>
                     <amp mcc>
                   <amp comment>'\n'
<amp allomorph> == \a' ' "<mor form>"' ' "<mor prop>"' '
                   "<mor sec>"' ' "<mor mec>"'\n'
<amp category> == \c' ' "<lex cat>"'\n'
<amp comment> == \co' ' "<lex com>"'\n'
<amp gloss morpheme name> == \g' ' "<lex gloss eng morname>"
 '\n'
<amp mor property> == \mp' ' "<lex prop>"'\n'
<amp noload> == \_no' ' "<lex noload>"'\n'
<amp order class> == \m' ' "<lex order class>"'\n'
<amp mcc> == \mcc' ' "<lex mcc>"'\n'
<amp root> == \r' '"<lex name>"'\n'
<amp underlying form> == \u' ' "<lex under>"'\n'.

```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%% Templates %%%%%%%%%%%%%%%%%%%%%%%%
%

ALLO1:
<lex form lex> == <lex form lex allo1>
<lex form lex allo1> == C_DLT.

ALLO2:
<lex form lex> == <lex form lex allo2>
<lex form lex allo2> == C_DLT.

ALLO3:
<lex form lex> == <lex form lex allo3>
<lex form lex allo3> == C_DLT.

ALLO4:
<lex form lex> == <lex form lex allo4>
<lex form lex allo4> == C_DLT.

ALLO5:
<lex form lex> == <lex form lex allo5>
<lex form lex allo5> == C_DLT.

ALLO6:
<lex form lex> == <lex form lex allo6>
<lex form lex allo6> == C_DLT.

ALLO7:
<lex form lex> == <lex form lex allo7>
<lex form lex allo7> == C_DLT.

ALLOMORPH:
<mor> ==
<mor form default> == "<phon under>".

DEFAULT:
<lex form lex> == <lex form lex default>
<lex form lex default> == C_DLT.

% Default Lexical Template
C_DLT:
<lex form lex> == I_AMPLE:<amp allomorph>.

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%% CATEGORIES  %%%%%%%%%%%%%%%%%%%%%%%%%
%
%

C_MORPHEME:
<> ==
<lex> ==
<amp lex> == I_AMPLE
```

```
<mor> == ALLOMORPH
<lex name> == "<phon under>"
<lex under> == "<phon under>"
<lex order class> ==
<lex form> == DEFAULT
<lex add prop> ==.

C_INFIX:
<> == C_MORPHEME
<lex type> == i.

C_PREFIX:
<> == C_MORPHEME
<lex type> == p.

C_ROOT:
<> == C_MORPHEME
<lex type> == r.

V09_10:
<> == C_ROOT
<lex cat> == 'VA'
<lex prop> == 'class2 ' "<lex add prop>".

V09:
<> == V09_10
<lex order> == 9.

V9i:
<> == V09
<lex prop> == 'vi ' V09:<lex prop>.

V9t:
<> == V09
<lex prop> == 'vt ' V09:<lex prop>.

V10:
<> == V09_10
<lex order> == 10
<lex prop> == V09_10:<lex prop> 'takes_null_S '.

V10i:
<> == V10
<lex prop> == 'vi ' V10:<lex prop>.

V10ic:
<> == V10
<lex prop> == V10i:<lex prop> 'takes_:_Caus'.

V10t:
<> == V10
<lex prop> == 'vt ' V10:<lex prop>.

%%%%%%%%%%%%%%%%%%%%%%%% YALALAG ZAPOTEC  %%%%%%%%%%%%%%%%%%%

R_chaa:
<> == V10I
```

```
<lex name> == 'cha\'a'
<lex noload> == '00129'
<phon under> == 'cha\''
<lex form> == DEFAULT ALLO1
<mor form allo1> == 'cha\'a   / _ #'
<mor sec default> == '/~_ #'
<lex gloss> == 'estar.en.grupo'
<lex add prop> == 'vt '
<lex mcc> == 'estar.en.grupo / ~Pl.pfx _'.

R_chach:
<> == V10i
<lex name> == 'chach'
<lex noload> == '00130'
<phon under> == 'cha:ch'
<lex gloss> == 'aparecer.muchos'
<lex com> == 'llach kwerp kello - effect of something that scares us'.

R_charall:
<> == V10i
<lex name> == 'cha\'ra\'ll'
<lex noload> == '00138'
<phon under> == 'cha\'la\'ch'
<lex gloss> == 'engordar'
<lex add prop> == 'takes_:_Caus '.

R_chel:
<> == V9i
<lex name> == 'chel'
<lex noload> == '00158'
<phon under> == 'che:l'
<lex gloss> == 'enrollar              | take group 2 pns?? '
<lex add prop> == 'takes_null_S takes_:_Caus'.

R_chez:
<> == V9i
<lex name> == 'che\'z'
<lex noload> == '00170'
<phon under> == 'che\'s'
<lex gloss> == 'romper'
<lex form> == ALLO1 ALLO2 DEFAULT
<mor form allo1> == 'che\'s   +/ [P-or-Modal]      _'
<mor form allo2> == 'e\'s     +/ H                _'
<mor mec default> == '+/ [P,H-or-Modal] ~_'
<lex add prop> == 'takes_:_Caus'.

R_chia:
<> == V9t
<lex noload> == '00172'
<phon under> == 'chia'
<lex gloss> == 'coser.vt'
<lex form> == ALLO1 DEFAULT
<mor form allo1> == 'chi  / _ [V-hi+cg]   +/ _ [1s-or-3r]'.

R_chio:
<> == V9i
```

```
<lex name> == 'chi\'o'
<lex noload> == '00195'
<phon under> == 'chi\'o'
<lex gloss> == 'crecer'
<lex form> == ALLO1 ALLO2 DEFAULT
<mor form allo1> == 'chi\'  / _ o\'  +/ _ 2s'
<mor form allo2> == 'chi\'kw / _ [V-hi+cg]   +/ _ [1s-or-3r]'
<lex add prop> == 'takes_:s_Caus'.

R_labj:
<> == V10i
<lex name> == 'labj'
<lex noload> == '00495'
<phon under> == 'lapj'
<lex gloss> == 'susmergir    | group 2 pns??'.

R_lall:
<> == V10i
<lex noload> == '00498'
<phon under> == 'la\'ll'
<lex gloss> == 'vagar'.

R_lap:
<> == V10t
<lex name> == 'lap'
<lex noload> == '00503'
<phon under> == 'la:p'
<lex gloss> == 'pizcar'.

R_len:
<> == V10i
<lex noload> == '00536'
<phon under> == 'len'
<lex gloss> == 'asociar'
<lex add prop> == 'takes_:__Caus '.

R_nit:
<> == V10i
<lex name> == 'nit'
<lex noload> == '00658'
<phon under> == 'li:t'
<lex gloss> == 'perder.vi'.

R_nnalÄllej:
<> == V10i
<lex name> == 'nna\'lÄllej'
<lex noload> == '00666'
<phon under> == 'na\':lchej'
<lex gloss> == 'sorprender'
<lex order> == '10  |no Habitual'.

R_nnao:
<> == V10t
<lex name> == 'nnao'
<lex noload> == '00667'
<phon under> == 'nao'
<lex form> == ALLO1 ALLO2 DEFAULT
```

```
<mor form allo1> == 'na  under_ao / _ o\'        +/ _ 2s'
<mor form allo2> == 'nakw         / _ [V-hi+cg] +/ _ [1s-or-3r]
                / _ [V-hi-cg]p  | where 1s or 3r is followed by 3f or 3a'
<lex gloss> == 'seguir'.

R_nne:
<> == V10t
<lex name> == 'nne'
<lex noload> == '00670'
<phon under> == 'ne'
<lex form> == ALLO1 ALLO2 DEFAULT
<mor form allo1> == 'ni / _ a\' +/ _ 1s'
<mor form allo2> == 'n, / _ # +/ [Modal] ... _
            +/ Rep ...     _
            +/ [Imperatives] ... _
            +/ Pl.pfx ... _'
<mor sec default> == '/ # ([C]) _  #
       /       _ ~#'
<lex gloss> == 'decir2'.

R_nnebia:
<> == V10
<lex name> == 'nnebia\''
<lex noload> == '00672'
<phon under> == 'nepia\''
<lex form> == ALLO1 DEFAULT
<mor form allo1> == 'ni / _ a\' +/ _ 1s'
<lex prop> == 'vt class2'                   % example of overriding an inherited value
<lex gloss> == 'gobernar'.

R_nnit:
<> == V10i
<lex name> == 'nni\'t'
<lex noload> == '00675'
<phon under> == 'ni\':t'
<lex gloss> == 'estar.regado | take group 2 pns?'.


R_nnitrao:
<> == V10t
<lex name> == 'nnitrao'
<lex noload> == '00676'
<phon under> == 'ni:tla'
<lex form> == ALLO1 ALLO2 DEFAULT
<mor form allo1> == 'ni:tla    under_ao  / _ o\'          +/ _ 2s'
<mor form allo2> == 'ni:tlakw            / _ [V-hi+cg]   +/ _ [1s-or-3r]'
<lex gloss> == 'borrar.vi'.

R_saa:
<> == V10i
<lex name> == 'sa\'a'
<lex noload> == '00752'
```

```
<phon under> == 'sa\''
<lex form> == ALLO1 ALLO2 ALLO3 DEFAULT
<mor form allo1> == 's   / _ a\'  +/ _ 1s'
<mor form allo2> == 'se\' / _ [V-hi-cg]   +/ _ [1s-or-3r]'
<mor form allo3> == 'sa\'a / _ #'
<lex sec default> == '/ ~_ #'
<lex add prop> == 'laryngealizes '
<lex gloss> == 'ir/salir/caminar   | caminar formerly separate record'.


R_sarall:
<> == V10i
<lex name> == 'sa\'ra\'ll'
<lex noload> == '00757'
<phon under> == 'sa\'la\'ch'
<lex gloss> == 'sospechar'
<lex com> == 'see also recordar2 wasa'ra'll'.

R_se:
<> == V10i
<lex noload> == '00759'
<phon under> == 'se'
<lex form> == ALLO1 ALLO2 ALLO3 DEFAULT
<mor form allo1> == 'si / _ a\' +/ _ 1s'
<mor form allo2> == 's, / _ # +/ [Modal] ... _
            +/ Rep ...     _
            +/ [Imperatives] ... _
            +/ Plural.S ... _'
<mor form allo3> == 's, / _ :tese| +/ [Modal] ... _
               | +/ Rep ...      _
               | +/ [Imperatives] ... _
               | +/ Plural.S ... _'
<lex sec default> == '/ # ([C]) _  #
      /         _ ~#'
<lex gloss> == 'parar.vi1'.

R_sella:
<> == V10i
<lex name> == 'sella'
<lex noload> == '00761'
<phon under> == 'secha'
<lex form> == ALLO1 ALLO2 ALLO3 DEFAULT ALLO4 ALLO5 ALLO6 ALLO7
<mor form allo1> == 'sech   / _ a\'  +/ _ 1s'
<mor form allo2> == 'seche   / _ e\'  +/ _ 3r'
<mor form allo3> == 'sech    / _ #'
<mor form allo4> == 'soch   / _ a\'  +/ _ 1s'
<mor form allo5> == 'soche   / _ e\'  +/ _ 3r'
<mor form allo6> == 'soch    / _ #'
<mor form allo7> == 'socha   / ~_ #'
<lex sec default> == '/ ~_ #'
<lex gloss> == 'parar.vi2 | Is -cha really part of root or is it an Adv? yes'.

R_serall:
<> == V10t
```

```
<lex name> == 'sera\'ll'
<lex noload> == '00764'
<phon under> == 'sela\'ch'
<lex gloss> == 'desear'.

R_sill:
<> == V10i
<lex name> == 'sill'
<lex noload> == '00773'
<phon under> == 'sich'
<lex gloss> == 'sonar'
<lex add prop> == 'takes_:_Caus '.

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A4.4 The Yalálag Zapotec DATR Query File

A4.4.1 Description

DATR theory (lexicon) files contain sets of nodes. Queries against a theory file may be made interactively from a DATR command line, or alternatively, may be run in batch mode. In batch mode, a query file may be used so the user does not have to manually type in each query. The Yalálag Zapotec query file listing in this appendix is used to generate AMPLE entries from the Yalálag Zapotec DATR theory file. Each entry in the query file has a corresponding entry in the original Yalálag Zapotec AMPLE root database file. The path `<amp lex mor name>` is used to direct the query to invoke the AMPLE interface and to use the morph name as the value for the record marker.

A4.4.2 File Listing

Filename: yal.qry

Note: line numbers added for reference only. Not in actual file.

```
R_chaa:<amp lex morname>
R_chach:<amp lex morname>
R_charall:<amp lex morname>
R_chel:<amp lex morname>
R_chez:<amp lex morname>
R_chia:<amp lex morname>
R_chio:<amp lex morname>
R_labj:<amp lex morname>
R_lall:<amp lex morname>
R_lap:<amp lex morname>
R_len:<amp lex morname>
R_nit:<amp lex morname>
R_nnalÄllej:<amp lex morname>
R_nnao:<amp lex morname>
R_nne:<amp lex morname>
R_nnebia:<amp lex morname>
R_nnit:<amp lex morname>
R_nnitrao:<amp lex morname>
R_saa:<amp lex morname>
R_sarall:<amp lex morname>
R_se:<amp lex morname>
R_sella:<amp lex morname>
R_serall:<amp lex morname>
R_sill:<amp lex morname>
```

A4.5 Batch File to Execute DATR

A4.5.1 Description

The batch file listed below is a standard DOS batch file. It contains two statements. The first calls the ZDATR program *zdatrtok*. Zdatrtok produces a token file from the yal.dtr file and names it yal.dtr.tok. The second program called is *zdatrinf* which is the inference program. This program uses both the *yal.qry* file and the *yal.dtr.tok* file as inputs, and then outputs the results of the query to a file named *yal.dtr.trc*. *yal.dtr.trc* is the Yalálag Zapotec trace file.

A4.5.2 File Listing

Filename: goyal.bat

Note: line numbers added for reference only. Not in actual file.

```
(12002)     zdatrtok -o yal.dtr.tok yal.dtr
(12003)     zdatrinf -j -u -V -1 -T yal.dtr.trc -Q yal.qry yal.dtr.tok
```

A4.6 The Yalálag Zapotec DATR Trace File

A4.6.1 Description

The result of running the batch file described above is a trace file called *Yal.dtr.trc*. Each line in the query file *Yal.qry* results in a corresponding line in the trace file. Note that although the trace file entries are close to AMPLE format, two things must be done before they are usable. First, empty lines and empty fields must be removed. Second, the trailing period must be stripped off. This is handled by a Perl script listed in Appendix Five.

A4.6.2 File Listing

Filename: yal.dtr.trc

Note: line numbers added for reference only. Not in actual file.

```
\r cha'a
\_no 00129
\u cha'
\a cha'  /~_ #
\a cha'a   / _ #
\c VA
\g estar.en.grupo
\m 10
\mp vi class2 vt takes_null_S
\mcc estar.en.grupo / ~Pl.pfx _
\co

.
\r chach
\_no 00130
\u cha:ch
\a cha:ch
\c VA
\g aparecer.muchos
\m 10
\mp vi class2 takes_null_S
\mcc
\co llach kwerp kello - effect of something that scares us

.
\r cha'ra'll
\_no 00138
\u cha'la'ch
\a cha'la'ch
\c VA
\g engordar
\m 10
\mp vi class2 takes_:_Caus takes_null_S
\mcc
\co

.
\r chel
\_no 00158
\u che:l
\a che:l
\c VA
\g enrollar             | take group 2 pns??
\m 9
\mp vi class2 takes_null_S takes_:_Caus
\mcc
\co
```

```
(12051)     .
(12052)     \r che'z
(12053)     \_no 00170
(12054)     \u che's
(12055)     \a che's    +/ [P-or-Modal]       _
(12056)     \a e's    +/ H                  _
(12057)     \a che's   +/ [P,H-or-Modal] ~_
(12058)     \c VA
(12059)     \g romper
(12060)     \m 9
(12061)     \mp vi class2 takes_:_Caus
(12062)     \mcc
(12063)     \co
(12064)
(12065)     .
(12066)     \r chia
(12067)     \_no 00172
(12068)     \u chia
(12069)     \a chi  / _ [V-hi+cg]   +/ _ [1s-or-3r]
(12070)     \a chia
(12071)     \c VA
(12072)     \g coser.vt
(12073)     \m 9
(12074)     \mp vt class2
(12075)     \mcc
(12076)     \co
(12077)
(12078)     .
(12079)     \r chi'o
(12080)     \_no 00195
(12081)     \u chi'o
(12082)     \a chi'  / _ o'  +/ _ 2s
(12083)     \a chi'kw / _ [V-hi+cg]   +/ _ [1s-or-3r]
(12084)     \a chi'o
(12085)     \c VA
(12086)     \g crecer
(12087)     \m 9
(12088)     \mp vi class2 takes_:s_Caus
(12089)     \mcc
(12090)     \co
(12091)
(12092)     .
(12093)     \r labj
(12094)     \_no 00495
(12095)     \u lapj
(12096)     \a lapj
(12097)     \c VA
(12098)     \g susmergir    | group 2 pns??
(12099)     \m 10
(12100)     \mp vi class2 takes_null_S
(12101)     \mcc
(12102)     \co
(12103)
(12104)     .
(12105)     \r la'll
(12106)     \_no 00498
(12107)     \u la'll
```

```
\a la'll
\c VA
\g vagar
\m 10
\mp vi class2 takes_null_S
\mcc
\co

.
\r lap
\_no 00503
\u la:p
\a la:p
\c VA
\g pizcar
\m 10
\mp vt class2 takes_null_S
\mcc
\co

.
\r len
\_no 00536
\u len
\a len
\c VA
\g asociar
\m 10
\mp vi class2 takes_:__Caus takes_null_S
\mcc
\co

.
\r nit
\_no 00658
\u li:t
\a li:t
\c VA
\g perder.vi
\m 10
\mp vi class2 takes_null_S
\mcc
\co

.
\r nna'lÄllej
\_no 00666
\u na':lchej
\a na':lchej
\c VA
\g sorprender
\m 10  |no Habitual
\mp vi class2 takes_null_S
\mcc
\co

.
```

```
(12165)     \r nnao
(12166)     \_no 00667
(12167)     \u nao
(12168)     \a na  under_ao / _ o'        +/ _ 2s
(12169)     \a nakw         / _ [V-hi+cg] +/ _ [1s-or-3r]
(12170)                     / _ [V-hi-cg]p  | where 1s or 3r is followed
   by 3f or 3a
(12171)     \a nao
(12172)     \c VA
(12173)     \g seguir
(12174)     \m 10
(12175)     \mp vt class2 takes_null_S
(12176)     \mcc
(12177)     \co
(12178)
(12179)     .
(12180)     \r nne
(12181)     \_no 00670
(12182)     \u ne
(12183)     \a ni / _ a' +/ _ 1s
(12184)     \a n, / _ # +/ [Modal] ... _
(12185)                 +/ Rep ...      _
(12186)                 +/ [Imperatives] ... _
(12187)                 +/ Pl.pfx ... _
(12188)     \a ne  / # ([C]) _  #
(12189)            /       _ ~#
(12190)     \c VA
(12191)     \g decir2
(12192)     \m 10
(12193)     \mp vt class2 takes_null_S
(12194)     \mcc
(12195)     \co
(12196)
(12197)     .
(12198)     \r nnebia'
(12199)     \_no 00672
(12200)     \u nepia'
(12201)     \a ni / _ a' +/ _ 1s
(12202)     \a nepia'
(12203)     \c VA
(12204)     \g gobernar
(12205)     \m 10
(12206)     \mp vt class2
(12207)     \mcc
(12208)     \co
(12209)
(12210)     .
(12211)     \r nni't
(12212)     \_no 00675
(12213)     \u ni':t
(12214)     \a ni':t
(12215)     \c VA
(12216)     \g estar.regado | take group 2 pns?
(12217)     \m 10
(12218)     \mp vi class2 takes_null_S
(12219)     \mcc
(12220)     \co
```

```

.
\r nnitrao
\_no 00676
\u ni:tla
\a ni:tla    under_ao  / _ o'          +/ _ 2s
\a ni:tlakw            / _ [V-hi+cg]   +/ _ [1s-or-3r]
\a ni:tla
\c VA
\g borrar.vi
\m 10
\mp vt class2 takes_null_S
\mcc
\co

.
\r sa'a
\_no 00752
\u sa'
\a s   / _ a'  +/ _ 1s
\a se' / _ [V-hi-cg]   +/ _ [1s-or-3r]
\a sa'a / _ #
\a sa'
\c VA
\g ir/salir/caminar   | caminar formerly separate record
\m 10
\mp vi class2 laryngealizes takes_null_S
\mcc
\co

.
\r sa'ra'll
\_no 00757
\u sa'la'ch
\a sa'la'ch
\c VA
\g sospechar
\m 10
\mp vi class2 takes_null_S
\mcc
\co see also recordar2 wasarall

.
\r se
\_no 00759
\u se
\a si / _ a' +/ _ 1s
\a s, / _ # +/ [Modal] ... _
            +/ Rep ...     _
            +/ [Imperatives] ... _
            +/ Plural.S ... _
\a s, / _ :tese| +/ [Modal] ... _
               | +/ Rep ...     _
               | +/ [Imperatives] ... _
               | +/ Plural.S ... _
\a se
\c VA
```

```
\g parar.vi1
\m 10
\mp vi class2 takes_null_S
\mcc
\co

.
\r sella
\_no 00761
\u secha
\a sech   / _ a'  +/ _ 1s
\a seche   / _ e'  +/ _ 3r
\a sech    / _ #
\a secha
\a soch   / _ a'  +/ _ 1s
\a soche   / _ e'  +/ _ 3r
\a soch    / _ #
\a socha   / ~_ #
\c VA
\g parar.vi2 | Is -cha really part of root or is it an Adv? yes
\m 10
\mp vi class2 takes_null_S
\mcc
\co

.
\r sera'll
\_no 00764
\u sela'ch
\a sela'ch
\c VA
\g desear
\m 10
\mp vt class2 takes_null_S
\mcc
\co

.
\r sill
\_no 00773
\u sich
\a sich
\c VA
\g sonar
\m 10
\mp vi class2 takes_:_Caus takes_null_S
\mcc
\co

.
```

APPENDIX 5

PERL SCRIPTS

A4.1 Script to Compare Database Files

A4.1.1 Description

The following Perl script serves two main purposes. First, it compares the Ogea AMPLE lexicon generated from DATR to the original Ogea AMPLE lexicon. Discrepencies are written to an error file. Second, it splits the generated lexicon into separate root and suffix database files. These separate files may be run directly within AMPLE.

The input files used for the comparison are the *Ogea.dtr.trc file* generated by DATR, and a file named *compare.txt*. Compare.txt is a file that merges the original Ogea AMPLE root database file and the suffix file into a single file. Each AMPLE record in compare.txt is a single line. Line breaks before all non-record marker fields were removed. Also, all whitespace was reduced to a single space character. In this way the original Ogea AMPLE lexicon was normalized for comparison to the DATR generated Ogea AMPLE lexicon. Comparing each record as a single line was easier programmatically than dealing with records where each field begins a new line.

A4.1.2 The File Listing

Filename: compare.pl.

Note: Line numbers not in original. Added for reference only.

```
# AUTHOR: Michael Colburn, mcolburn@sprintmail.com
# DATE:  November, 1999
# Filename: Compare.pl
# Purpose:
#    1.  Cleans up the output of Ogea DATR lexicon queries for
#        comparison to the original AMPLE verion,
#    2.  Writes discrepencies to an error file,
#    3.  Writes cleaned up version to an AMPLE unified
#        database file.

# original.db is the original Ogea AMPLE unified database file
open (AMPLE_IN,
        'c:\program files\sil\carlastudio\ogea\original.db');

# ogea.dtr.trc is the DATR trace (output) file
#     resulting from running ogea.qry.
open (DATR_IN, 'c:\zdatr\zdatr20\program\bin\ogea.dtr.trc');
open (ERROR_OUT,
">c:\\zdatr\\zdatr20\\program\\bin\\error.txt");

# Output creates an AMPLE unified database file.
open (AMPLE_OUT,
      ">c:\\program files\\sil\\carlastudio\\ogea\\ogea.db");

$Error_CNT = 0;
$/ = "\n\n";  # set record delimiter to two line breaks

while (<AMPLE_IN>) {

  # Normalize the AMPLE input record
  $AMPL = $_;
  $AMPL =~ s/\n/ /g; # replace line breaks with spaces
  $AMPL =~ s/  / /g; # get rid of extra spaces
  $AMPL =~ s/  / /g; # get rid of extra spaces
  $AMPL =~ s/  / /g; # get rid of extra spaces

  # Normalize the DATR input record
  $DATR = <DATR_IN>;
  $DATR =~ s/^\.\n//g;  # get rid of period inserted by DATR

  # get rid of empty fields
  $DATR =~ s/\\\w\w\w \\/\\/g;
  $DATR =~ s/\\\w\w\w \n/\n/g;
  $DATR =~ s/\\\w\w \\/\\/g;
  $DATR =~ s/\\\w\w \n/\n/g;
  $DATR =~ s/\\\w \\/\\/g;
  $DATR =~ s/\\\w \n/\n/g;

  $DATR =~ s/\n/ /g; # replace line breaks with spaces
  $DATR =~ s/  / /g; # get rid of extra spaces
  $DATR =~ s/  / /g; # get rid of extra spaces
  $DATR =~ s/  / /g; # get rid of extra spaces

  # The following code compares the DATR output to the
  # original AMPLE files and writes out lines that
  # fail the comparison

  $DATR_CNT++;
  if (($AMPL ne $DATR) && ($DATR ne '')) {
    print ERROR_OUT "A:$AMPL"."\nD:$DATR\n";
    $Error_CNT++;
  }

  # put line breaks back in
  $DATR =~ s/\\/\n\\/g;
  print AMPLE_OUT "$DATR\n";
}

close (AMPLE_IN);
close (DATR_IN);
close (ERROR_OUT);
close (AMPLE_OUT);
print "AMPLE: $AMPL_CNT DATR: $DATR_CNT Errors: $Error_CNT\n";
```

A4.2 Script to Generate AMPLE Database File

A4.2.1 Description

The following Perl Script is a version of the previous script (compare.pl) that simply cleans up the DATR trace file and writes out an AMPLE dictionary database file.

A4.2.2 The File Listing

Filename: DToAMPLE.pl.

Note: Line numbers not in original. Added for reference purposes.

```
# AUTHOR: Michael Colburn, mcolburn@sprintmail.com
# DATE:  November, 1999
# Filename: DToAMPLE.pl
# Purpose: Cleans up DATR trace file, writes to AMPLE
#          unified database file.

# ogea.dtr.trc is the DATR trace (output) file
# resulting from running ogea.qry.
open (DATR_IN,
        'c:\zdatr\zdatr20\program\bin\ogea.dtr.trc');

# Output creates an AMPLE unified database file
open (AMPLE_OUT,
  ">c:\\program files\\sil\\carlastudio\\ogea\\ogea.db");

$/ = "\.\n"; # set record delimiter to match DATR's output
```

```
while (<DATR_IN>) {

  $DATR = $_;

  $DATR =~ s/\.\n//g;  # get rid of period inserted by DATR

  # get rid of empty fields
  $DATR =~ s/\\\w\w\w \\/\\/g;
  $DATR =~ s/\\\w\w\w \n/\n/g;
  $DATR =~ s/\\\w\w \\/\\/g;
  $DATR =~ s/\\\w\w \n/\n/g;
  $DATR =~ s/\\\w \\/\\/g;
  $DATR =~ s/\\\w \n/\n/g;


  $DATR =~ s/\n/ /g; # replace line breaks with spaces
  $DATR =~ s/  / /g; # get rid of extra spaces
  $DATR =~ s/  / /g; # get rid of extra spaces
  $DATR =~ s/  / /g; # get rid of extra spaces

  # put line breaks back in
  $DATR =~ s/\\/\n\\/g;
  print AMPLE_OUT "$DATR\n";
}

close (AMPLE_IN);
close (AMPLE_OUT);
```

A4.3 Scripts to Compare AMPLE Output

A4.3.1 Description

This Perl script compares two AMPLE interlinear files. The interlinear files are the result of running AMPLE in CARLA Studio, using the file list.txt as the text file to be parsed. First AMPLE is run using the hand-created Ogea AMPLE database file *original.db*, which is copied to *ogea.db* for the run. The resulting interlinear file, list.itx is renamed (by hand) to *old.itx*. Next, the DATR to AMPLE interface is executed using the batch file *go.bat*, which creates a new version of *ogea.db*, generated from DATR (see Appendix 3 for the listing). AMPLE is run again, and this time the resulting interlinear file is named *new.itx*. Finally, the Perl script *C2.pl* is run.

A4.3.2 The File Listing

Filename: C2.pl.

Note: Line numbers not in original. Added for reference purposes.

```
(12442)     # AUTHOR: Michael Colburn, mcolburn@sprintmail.com
(12443)     # DATE:  November, 1999
(12444)     # Filename: C2.pl
(12445)     # Purpose: compares the AMPLE output from using the hand
(12446)     #          developed Ogea AMPLE database file to the
(12447)     #          AMPLE output from using the version generated
(12448)     #          through DATR
(12449)
(12450)     # original.itx is the original Ogea AMPLE output file
(12451)     open (OLD_IN, 'c:\program files\sil\carlastudio\ogea\ERI
(12452)                    Interlinear Files\original.itx');
(12453)
(12454)     # new.itx is the interlinear file AMPLE produces using the
(12455)     # database file generated from DATR
(12456)     open (OLD_IN, 'c:\program files\sil\carlastudio\ogea\ERI
(12457)                    Interlinear Files\new.itx');
(12458)
(12459)     open (ERROR_OUT,
(12460)                    ">c:\\program
   files\\sil\\carlastudio\\ogea\\ERI
(12461)                    Interlinear Files\\error.txt");
(12462)
(12463)     $Error_CNT = 0;
(12464)
(12465)     while (<OLD_IN>) {
(12466)       $OLD_CNT++;
(12467)       $OLD = $_;
(12468)       $NEW = <NEW_IN>;
(12469)       $NEW_CNT++;
(12470)       if (($OLD ne $NEW) && ($NEW ne '')) {
(12471)         print ERROR_OUT "O:$OLD"."N:$NEW\n";
(12472)         $Error_CNT++;
(12473)       }
(12474)     }
(12475)
(12476)     close (OLD_IN);
(12477)     close (NEW_IN);
(12478)     close (ERROR_OUT);
(12479)     print "OLD: $OLD_CNT NEW: $NEW_CNT Errors: $Error_CNT\n";
```

A4.4 Scripts to Calculate Reduction of Redundancy

A4.4.1 For Ogea

A4.4.1.1 The AMPLE File

A4.4.1.1.1 Description

The Perl script *AMPStat.pl* examines each line of each record in the hand-built Ogea AMPLE database file *original.db*. A counter variable exists for each field type found in a record. The value of each counter is displayed after the entire file has been processed. These values are compared to the output of the file *DATStat.pl*, listed in another section below.

A4.4.1.1.2 The File Listing

Filename: AMPStat.pl.

Note: Line numbers not in original. Added for reference purposes.

```
# AUTHOR: Michael Colburn, mcolburn@sprintmail.com
# DATE:  November, 1999
# Filename: AMPStat.pl
# Purpose: Counts the number of occurrences of
#          specified fields.  This number is compared to
#          the number of occurrences
#          of the corresponding DATR paths in the
#          Ogea DATR lexicon.  This provides a measure of the
#          reduction of redundancy by using DATR.


open (AMPLE_IN, 'c:\program
                files\sil\carlastudio\ogea\original.db');
```

```
while (<AMPLE_IN>) {
    $af++  if (/\\a/);  # allomorph
    $sec++ if (/\/ /);  # SEC
    $mec++ if (/\+\//); # MEX
    $c++   if (/\\c/);  # Category
    $co++  if (/\\co/); # Comment
    $f++   if (/\\f/);  # Feature
    if (/\\g/) {        # Gloss
      if (/\\ge/) {     # English gloss
        $ge++;
      } else {
        $g++;           # Morphname
      }
    }
    $u++    if (/\\u/);    # underlying form
    $gn++   if (/\\gn/);   # national language
    $loc++  if (/\\loc/);  # infix location
    $mp++   if (/\\mp/);   # morpheme property
    $o++    if (/\\o/) ;   # order class
    $mcc++  if (/\\mcc/);  # MCC
    $type++ if (/\\type/); # record type
}

print "\n af:$af \n ap:$ap \n sec:$sec \n mec:$mec \n c:$c
       \n co:$co \n f:$f \n ge:$ge \n u:$u \n gn:$gn
       \n loc:$loc
       \n g:$g \n mp:$mp \n o:$o \n mcc:$mcc \n type:$type\n";
close (DATR_IN);
```

A4.4.1.2 The DATR Ffile

A4.4.1.2.1 Description

The Perl script *DATStat.pl* examines each path of each root node in the Ogea DATR theory file *Ogea.dtr*. A counter variable exists for each path type found in a node. The value of each counter is displayed after the entire file has been processed. These values are compared to the output of the file *AMPStat.pl*, listed above.

A4.4.1.2.2 The File Listing

Filename: DATStat.pl.

Note: Line numbers not in original. Added for reference purposes.

```
# AUTHOR: Michael Colburn, mcolburn@sprintmail.com
# DATE:  November, 1999
# Filename: DATStat.pl
# Purpose: Counts the number of occurrences of specified paths
#          This number is compared to the number of
#          occurrences of the corresponding AMPLE field
#          markers in the Ogea AMPLE lexicon.  This provides
#          a measure of the reduction of redundancy
#          by using DATR.

open (DATR_IN, 'c:\zdatr\zdatr20\program\bin\ogea.dtr');

while (<DATR_IN>) {
  if (/^%/) {
    next;
  }
  if (/^ENG_/) {
    $Hit_ENG = 1;
    $ECNT++;
  }
  if ($Hit_ENG) {
    $ge++ if (/^<lex gloss eng>/);
    $g++  if (/^<lex gloss eng morname>/);
  }
  if (/^R_/ || /^S_/) {
    $Hit_ENG = 0;
    $Hit_R = 1;
    $RCNT++;
  }
  if ($Hit_R) {
    $af++   if (/^<allo form/);
    $af++   if (/^<lex form/);
    $ap++   if (/^<allo prop/);
    $sec++  if (/^<allo sec/);
    $mec++  if (/^<allo mec/);
    $c++    if (/^<lex cat/);
    $co++   if (/^<lex com/);
    $f++    if (/^<lex feat/);
    $u++    if (/^<phon under/);
    $gn++   if (/^<lex gloss nat/);
    $loc++  if (/^<lex loc/);
    $mp++   if (/^<lex prop/);
    $o++    if (/^<lex order class/);
    $mcc++  if (/^<lex mcc/);
    $type++ if (/^<lex type/);
  }
}

print "\nRoot or Suffix Nodes: $RCNT \n ENG_ nodes: $ECNT
 \n af:$af \n ap:$ap \n sec:$sec \n mec:$mec \n c:$c
 \n co:$co \n f:$f \n ge:$ge \n u:$u \n gn:$gn \n loc:$loc
 \n g:$g \n mp:$mp \n o:$o \n mcc:$mcc \n type:$type\n";
close (DATR_IN);
```

A4.4.2 For Yalalog Zapotec

A4.4.2.1 The AMPLE File

A4.4.2.1.1 Description

The Perl script *AMPStat2.pl* examines each line of each record in the Yalalog Zapotec AMPLE database file `yal0910`*.db*. This file is an extraction from the file *yalrt.db*, and contains all Order Class 10 records, and four Order Class 9 records. In the script, a counter variable exists for each field type found in a record. The value of each counter is displayed after the entire file has been processed. These values are compared to the output of the file *DATStat2.pl*, listed in another section below.

A4.4.2.1.2 The File Listing

Filename: AMPStat2.pl.

Note: Line numbers not in original. Added for reference purposes.

```
# AUTHOR: Michael Colburn, mcolburn@sprintmail.com
# DATE:  November, 1999
# Filename: AMPStat2.pl
# Purpose: Counts the number of occurrences of
#          specified fields.  This number is compared
#          to the number of occurrences
#          of the corresponding DATR paths in the
#          Yalalog Zapotec DATR lexicon.  This provides
#          a measure of the reduction of redundancy
#          by using DATR.
```

```
open (AMPLE_IN, 'c:\zdatr\zdatr20\program\bin\yal0910.db');
while (<AMPLE_IN>) {

    $af++  if (/\\a /);  # allomorph
    $sec++ if (/\/ /);   # SEC
    $mec++ if (/\+\//);  # MEX
    $c++   if (/\\c /);  # Category
    $co++  if (/\\co /); # Comment
    $f++   if (/\\f /);  # Feature
    if (/\\g /) {        # Gloss
      if (/\\ge /) {     # English gloss
        $ge++;
      } else {
        $g++;            # Morphname
      }
    }
    $no++   if (/\\_no /);  # no load
    $u++    if (/\\u /);    # underlying form
    $gn++   if (/\\gn /);   # national language
    $loc++  if (/\\loc /);  # infix location
    $mp++   if (/\\mp /);   # morpheme property
    $o++    if (/\\m /) ;   # order class
    $mcc++  if (/\\mcc /);  # MCC
    $type++ if (/\\type /); # record type
}

print "\n af:$af \n ap:$ap \n sec:$sec \n mec:$mec
       \n No load:$no \n c:$c \n co:$co \n f:$f \n ge:$ge
       \n u:$u \n gn:$gn \n loc:$loc \n g:$g \n mp:$mp
       \n o:$o \n mcc:$mcc \n type:$type\n";
close (DATR_IN);
```

A4.4.2.2 The DATR File

A4.4.2.2.1 Description

The Perl script *DATStat2.pl* examines each path of each root node in the Yalalog Zapotec DATR theory file *yal.dtr*. A counter variable exists for each path type found in a node. The value of each counter is displayed after the entire file has been processed. These values are compared to the output of the file *AMPStat2.pl*, listed above.

A4.4.2.2.2 The File Listing

Filename: DATStat2.pl.

Note: Line numbers not in original. Added for reference purposes.

```
# AUTHOR: Michael Colburn, mcolburn@sprintmail.com
# DATE:  November, 1999
# Filename: DATStat2.pl
# Purpose: Counts the number of occurrences of specified paths
#          This number is compared to the number of
#          occurrences
#          of the corresponding AMPLE field markers in the
#          Yalalog Zapotec AMPLE lexicon.  This provides a
#          measure of the reduction of redundancy by
#          using DATR.
open (DATR_IN, 'c:\zdatr\zdatr20\program\bin\yal.dtr');
while (<DATR_IN>) {
if (/^%/) {
    next;
}
  if (/^R_/) {
    $Hit_R = 1;
    $RCNT++;
  }
  if ($Hit_R) {
    $af++   if (/^<allo form/ || (/^<mor form/));
    $ap++   if (/^<allo prop/);
    $c++    if (/^<lex cat/);
    $co++   if (/^<lex com/);
    $f++    if (/^<lex feat/);
    $ge++   if (/^<lex gloss>/);
    $gn++   if (/^<lex gloss nat/);
    $loc++  if (/^<lex loc/);
    $mcc++  if (/^<lex mcc/);
    $mec++  if (/^<mor mec/);
    $mp++   if (/^<lex prop/ || /^<lex add prop/);
    $name++ if (/^<lex name/);
    $no ++  if (/^<lex noload>/);
    $o++    if (/^<lex order/);
    $sec++  if (/^<mor sec/);
    $type++ if (/^<lex type/);
    $u++    if (/^<phon under/);
  }
}

print "\nRoot Nodes: $RCNT \n Names:$name \n No load:$no
       \n af:$af \n ap:$ap \n sec:$sec \n mec:$mec
       \n c:$c \n co:$co \n f:$f \n ge:$ge \n u:$u
       \n gn:$gn \n loc:$loc \n g:$g \n mp:$mp \n o:$o
       \n mcc:$mcc \n type:$type\n";
close (DATR_IN);
```