uruwi

IIIIIIIIIIIIIII

aaaaaaaaaaaaaaaa

A complete grammar

20 December 2017

Dedicated to Mareck.

```
Branch: canon
Version: 0.1
Date: 2017-11-12 (28 ruj fav)
```

Contents

0.1 | Introduction

1 | Phonology and orthography

1.1 | Phoneme inventory

Synopsis: *Consonants are in free variation with vowels.*

In aaaaaaaaaaA, each consonant is interchangeable with a corresponding vowel. Consonants may also have an ingressive pronuniation.

Table 1.1: Phonemes of aaaaaaaaaaA.

	Consonant			
#	Hacm	(Egre)	(Ingr)	Vowel
0	ɾ	tʼ	ǃ	e˩
1	nᵠ	ŋ	ɠ	ã
2	ɓ	p	p	y
3	ɒ	m	ɓ	ũ
4	ɻ	s	s	ɨ
5	ʮ	j	j	i
6	ɭ	tɬʼ	‖	ʌ˩
7	o	w	w	u
8	ɸ	k	k	o
9	ǀ	l	l	ɯ
10	ɬ	kʼ	ǂ	o˩
11	ɲ	n	n	ẽ
12	ɗ	r	ɹ	e̞
13	ɑ	t	t	e
14	h	ħ	h	a
15	ʂ	ɬ	ɬ	ʌ

When pronounced ingressively, the tones of vowels are inverted. That is, [ʌ˩↑] becomes [ʌ˥↓]. Nasal vowels also gain a characteristic hissing sound from air entering through the nose.

1.2 | Airflow

Synopsis: *Change of airflow direction has a morphosyntactic basis.*

There are two types of airflow: *ingressive* and *egressive*. The direction of airflow is reversed:

- at the beginning of a descriptor
- at certain affixes
- in the middle of certain roots

On a proper noun, the direction is switched to egressive and remains so until it is changed by one of the above methods.

In hacm, switching the direction of airflow is marked by </> (to ingressive) and <\> (to egressive). In dictionaries, a switch in airflow direction (without regard to the final state) is marked using <>>.

1.3 | Phonotactics

The only phonotactic restriction is that two identical instances of a phoneme may not occur consecutively. If this rule is violated by affixation, then the violation is resolved by:

- replacing the earlier instance with an instance of its predecessor (e. g. /w/ (7) → /tɬ'/ (6), wrapping when necessary), and
- replacing the later instance with an instance of its successor (e. g. /w/ (7) → /k/ (8), wrapping when necessary).

1.4 | Allophony

The following changes are made:

$$\begin{aligned} \text{lm} &\to \text{p} \\ \text{nl} &\to \text{r} \\ \text{ɬ} &\to \text{t'} \quad (\blacklozenge\neg\{\square, \text{t'}, \text{k'}\}) \end{aligned}$$

(Here, the symbols for the egressive versions of the consonants are used, but these rules apply during ingressive airflow as well.)

Thus, for instance, /ħswlmŋ/ would be resolved to [ħswpŋ], which could, for instance, be pronounced [asupã].

1.5 | The biting affix

A frequent type of affix encountered in aaaaaaaaaA is the *biting affix*, which has the syntax <–>δ:ω>. To apply this affix onto a word α:

- Take the last length(δ) phonemes of α, and xor them with δ itself using the indices of the phonemes.
- In addition, switch the airflow at the start of the altered phonemes.
- Then append ω.
- Resolve any phonotactic violations.

For instance, if we wanted to use ⟨–›ɥːʃn⟩ on ⟨ɒɷl⟩, then we would:

- Take the last letter of ⟨ɒɷl⟩, namely ⟨–l⟩ (9), and xor it with ⟨–ɥ⟩ (5). $9 \veebar 5 = 12$ so we now have ⟨–ɳ⟩.
- Append ⟨–ʃn⟩. We now have ⟨ɒɷ›ɳʃn⟩.

2 | The statement space

2.1 | Conceptualisation

Synopsis: *aaaaaaaaaA makes a distinction not between nouns and verbs, but rather between concretes and abstracts.*

Table 2.1: Distinction between concretes and abstracts.

Concretes	Abstracts
Describe concrete objects and actions	Describe abstract concepts, processes and relations
Inflected for gender and number mutability	Not inflected
Mutual order in parameter list is usually significant	Mutual order in parameter list is insignificant

Thus, if $C_1, C_2, \ldots, C_n$ are concretes, $A_1, A_2, \ldots, A_n$ are abstracts, and X is either a concrete or an abstract, then

$$X(A_1, \ldots, A_n, C_1, \ldots, C_n) \tag{2.1}$$

means that X has the properties $A_1, \ldots, A_n$ and involves $C_1, \ldots, C_n$.

For instance, take the sentence *The sun shines.* This can be translated to Source(Sun, Light). In this case, Source is an abstract, and Sun and Light are concretes. Literally, the translation says that the sun and light are involved in sourcing, or the sun is a source of light.

As a more complex example, *On a Sunny morning after the [summer] solstice we started for the mountains* can be translated as:

```
Time(
 Morning(Weather(Sun), After(Summer_Solstice)),
 Walk(We, Destination(Mountain), Start))
```

2.2 | Application

The top level of the statement tree is treated differently from the lower levels. The syntax of the top level is

$$\text{Topic Operator Arguments} \ldots \equiv \text{Operator}(\text{Topic}, \text{Arguments}) \tag{2.2}$$

$$\text{Topic Operator Args}_1 \triangle \text{Args}_2 \ldots \equiv \text{Operator}(\text{Args}_1, \text{Topic}, \text{Args}_2) \tag{2.3}$$

$$\blacktriangleright \text{ Operator Arguments} \ldots \equiv \text{Operator}(\text{Arguments}) \tag{2.4}$$

Note that in (2.2) and (2.4), all of the components of the syntax can be concretes or abstracts. In (2.3), Topic must be a concrete, but all other arguments may be concretes or abstracts.

The lower levels use the following syntax:

$$X\ulcorner \text{ Arguments } \ldots \lrcorner \equiv X(\text{Arguments}) \tag{2.5}$$

Inside the topic, the following is used instead:

$$\llcorner \text{ Arguments } \ldots X\urcorner \equiv X(\text{Arguments}) \tag{2.6}$$

At the end of the sentence, any number of ⌟s can be omitted.

Finally, here are the morphemes that aaaaaaaaaA assigns to the special symbols:

Table 2.2: Names of syntactic markers in aaaaaaaaaA.

△	ɟs
▶	h
⌜	–>ɥːɾn
⌟	b
⌞	ɸ
⌝	–>ɾːbh

2.3 | Concrete inflections

2.3.1 | Season class

Synopsis: *There are five classes open to new concretes, as well as a closed class of season-neutral words.*

In general, if C is of class y, then the processed form of C will be of class $y + 1$ (or 1 if $y = 5$).

Table 2.3: Classes in aaaaaaaaaA.

#	(° from VE) Start	End	Name	Archetypes
1	24	96	Late Spring / Early Summer	decorative flora such as flowers, honey, bees
2	96	168	Late Summer / Early Autumn	raw plant-based crops, milk, trees, grass, hot things, rain, most aquatic creatures and insects

#	(° from VE) Start	End	Name	Archetypes
3	168	240	Mid Autumn / Early Winter	processed plant-based food, wood
4	240	312	Mid Winter	(meat of) wild animals, frozen or cold things, metals
5	312	24	Late Winter / Early Spring	decorative flora such as flowers, arachnids

2.3.2 | Number mutability

Number mutability describes how likely the quantity of a concrete is to change. Note that a concrete can only take either a time mutability or a space mutability, not both.

Table 2.4: List of number mutabilities.

Symbol	Name	Description
Ω	Multiversal time-constant	The quantity cannot change *under any circumstances*, or quantity is meaningless or irrelevant in this context.
Ψ	Universal time-constant	The quantity does not change within the current universe, but might be different in other universes.
X	Lifetime-constant	The quantity is unlikely to change to a significant degree within one's lifetime.
Z	Lifetime-enumerable	The quantity is likely to change one or more times during one's lifetime, but such a change would be a significant life event.
Ξ	Continually mutable	The quantity is likely to change within a short time span (usually within seconds or minutes, but can be up to about a month).
Φ	Continuously mutable	The quantity is continuously changing across time.
$\vec{\Omega}$	Multiversal space-constant	The quantity is currently the same across all universes.
$\vec{\Psi}$	Universal space-constant	The quantity is currently the same within the current universe, but might be different in other universes.
$\vec{X}$	Domain space-constant	The quantity is unlikely to be different within the current domain.
$\vec{\Xi}$	Continually space-mutable	The quantity is likely to change across a short span of space (usually a few metres, but can exceed hundreds of kilometres).
$\vec{\Phi}$	Continuously space-mutable	The quantity is continuously changing across space.
Σ	Situational	(in programming) The quantity might depend on the implementation.

Table 2.5: List of number mutability affixes.

S#	Affix 1
1	–>nᵠ:
2	–>ɟ:
3	–>l:
4	–>b:
5	–>n:
0	–>h:

Sym	Affix 2
Ω	lbɒ
Ψ	lɑnᵠ
X	bɑɟ
Z	lʃɥ
Ξ	ɒɥɒ
Φ	ɑsl
$\vec{\Omega}$	lɒɥ
$\vec{\Psi}$	lhɒ
$\vec{X}$	bhl
$\vec{\Xi}$	ɒlɥ
$\vec{\Phi}$	ɑʃɾd
Σ	lnᵠo

When a concrete acts as a verb, it inherits the number mutability of its first argument.

If the mutability equals that of the previous concrete mentioned in the same sentence, then both the class affix and the mutability affix can be omitted.

2.4 | A simple example

Take the sentence *Fish eat flowers*, which would be treed into Eat(Fish, Flower).

The roots we need are:

- ⟨boɸ⟩ *c2* (0) eats (1)
- ⟨ɸnlʃ⟩ *c2* fish
- ⟨ɑsn⟩ *c1* flower

The number of fish that exist change whenever a fish is born or dies. This is quasi-continuous, but technically continual. We can choose either option but we will use Ξ in this example. We use the same mutability for flowers.

Fronting the topic, we get:

\ɸnl/ɟɒɥɒ bo\ɾdɒɥɒ ɑs/lɒɥɒ
ɸnlʃ–>ɟ:ɒɥɒ boɸ–>ɟ:ɒɥɒ ɑsn–>nᵠ:ɒɥɒ
fish-2-Ξ eat-2-Ξ flower-1-Ξ
Fish eat flowers.

However, we can omit the affixes on all but the first word, leaving:

\ɸnl/ɟɒɥɒ boɸ ɑsn
ɸnlʃ–>ɟ:ɒɥɒ boɸ ɑsn
fish-2-Ξ eat flower
Fish eat flowers.

Now take the earlier sentence *On a Sunny morning after the [summer] solstice we started for the mountains*, whose tree representation is:

```
Time(
  Morning(Weather(Sun), After(Summer_Solstice)),
  Walk(We, Destination(Mountain), Start))
```

Morning(Weather(Sun), After(Summer_Solstice)) (in topic position) can be translated as:

```
\φ φ ſo/lbɑȷ ɒφȷ\nᵠɑs/lbh φ ɭbφ\\lbɑȷ ȷ/ɑbh φl\nᵠɒh
φ φ ſφl–>nᵠ:bɑȷ ɒφȷs–>h:ɑsɬ–>ɾl:bh φ ɭblɑ–>ȷ:bɑȷ ȷnᵠ–>ɾl:bh φlh–>ɾl:bh
└ └ sun-1-X⃗ weather-0-Φ-┐└ summer_solstice-2-X⃗ after-┐morning(X⃗)-┐
```

The rest of the sentence is thus:

```
ȷφɒ ȷɑ/bɒʏ\\lſn ɭlɑ ns/nᵠn ɾlΛφɒlʏ b ɒll
ȷφɒ ȷɑɒ–>nᵠ:ɒʏɒ–>ʏ:ſn ɭlɑ nʏ–>ʏ:ſn ɾlſɒ–>n:ɒlʏ b ɒll
time walk-1-Ξ-┌ speaker(Ξ) destination-┌ mountain-5-Ξ⃗ ┘start
```

2.5 | Modifiers

Synopsis: *Modifiers can be divided into two categories: descriptors and quantifiers.*

Table 2.6: Distinction between descriptors and quantifiers.

Descriptors	Quantifiers
General category	Modifiers such as “every”, “some” and “most” that signify a relationship
Open class (derived from concretes and abstracts)	Closed class
Follow the separation rule	Do not follow the separation rule

2.5.1 | Descriptors

Semantically, descriptors act like expression trees that are covered by their antecedents. For instance, if Weather was modified by a descriptor acting like Sun, then the resulting tree would be Weather(Sun).

Descriptors can modify only proper expression trees below the top level and not other descriptors.

Descriptors follow the separation rule, which states that:

- A descriptor must fall somewhere after what it modifies.
- A descriptor may not be adjacent to what it modifies, or to any other descriptor modifying the same antecedent.
- A descriptor must fall as early as possible under the above two rules.
- Given D_1 and D_2 which can both occupy a certain position, D_1 is prioritised before D_2 if the antecedent of D_1 falls before that of D_2.

This yields the following algorithm for getting the next word:

- If there are no eligible outstanding descriptors, then print the next non-descriptor word.
- If there are any eligible outstanding descriptors, then print the one whose antecedent falls the earliest and remove it from the list of outstanding descriptors.

The archetypal form of the descriptor is a straight derivation from an abstract or a concrete whose expression tree is the same word. This avoids the ⌜⌟-overhead that usually applies. In this form, the direction of airflow is switched at the beginning of the descriptor:

⟨ɸlh⟩ Morning → ⟨>ɸlh⟩ D(Morning)
⟨ɟɸɒ⟩ Time → ⟨>ɟɸɒ⟩ D(Time)

Note that descriptors are not inflected, even if they come from concretes.
Thus the previous example can also be translated as:

\ɸ ɒɸɟ/nᵠɑsl ɟnᵠ \ɾɸl /lblɑ ɸl\nᵠɒh
ɸ ɒɸɟs–>h:ɑsl ɟnᵠ >ɾɸl >lblɑ ɸlh–>ɾ:bh
⌞ weather-0-Φ after D-sun D-summer_solstice morning-⌝

ɟɸɒ ɟɑ/bɒɥ\lɾn nɥ /llɑ \ɾllɒ /ɒl
ɟɸɒ ɟɑɒ–>nᵠ:ɒɥɒ–>ɥ:ɾn nɥ >llɑ >ɾllɒ >ɒl
time walk-1-Ξ-⌜ destination D-speaker D-mountain D-start

Other descriptors are possible:

Table 2.7: Other descriptors.

Prefix	Input	Output
>ɾ'	*C*	Inalienable_Possession(*C*)
>ɾɑ'	*C*	Association(*C*)
>ɸb'	*C*	Property(*C*)
>ll'	*C*	Borrow(*C*)
>ɟh'	*C*	Destination(*C*)

2.5.2 | Quantifiers

Quantifiers narrow their antecedents, and include words such as *all* or *some*:

$$\text{All } X\text{s are } Y\text{s} \equiv X \subseteq Y \tag{2.7}$$
$$\text{Some } X\text{s are } Y\text{s} \equiv (X \cap Y) \neq \varnothing \tag{2.8}$$
$$\text{No } X\text{s are } Y\text{s} \equiv (X \cap Y) = \varnothing \tag{2.9}$$

Table 2.8 lists the *conservative* quantifiers of aaaaaaaaaaA. These quantifiers satisfy $Q(X, Y) \iff Q(X, X \cap Y)$. Quantifiers where this is not the case are listed in table 2.9.

Unlike descriptors, quantifiers are not subject to the separation rule. In fact, *they must immediately follow what they quantify, even if doing so means that a descriptor must be delayed.* This means that the algorithm in subsection 2.5.1 must be modified to read as such:

Table 2.8: Conservative quantifiers.

Quantifier (Q)	Translation	Meaning of *Q Xes are Y* ($Q(X, Y)$)
⟩hɥ	All	$X \subseteq Y$
ɥnᵠl	Not all	$X \not\subseteq Y$
ɑɥɸ	Some	$(X \cap Y) \neq \varnothing$
b⟩o	None	$(X \cap Y) = \varnothing$
ɒɥo	Most	$\lvert X \cap Y\rvert \geq \lvert X - Y\rvert$
ɸ⟩h	At least two	$\lvert X \cap Y\rvert \geq 2$
onɸ	One	$\lvert X \cap Y\rvert = 1$
ɒnnᵠ	Half of	$\lvert\lvert X \cap Y\rvert - \lvert X - Y\rvert\rvert \leq 1$
b⟩l	A finite number of (sometimes "many")	$\lvert X \cap Y\rvert < \aleph_0$
sjh	A countable number of	$\lvert X \cap Y\rvert \leq \aleph_0$
ɑ⟩l	An infinite number of	$\lvert X \cap Y\rvert \geq \aleph_0$
ſnnᵠ	An uncountable number of	$\lvert X \cap Y\rvert > \aleph_0$
⟩jl	Almost all	$\lvert X - Y\rvert < \aleph_0 \wedge \lvert Y\rvert \geq \aleph_0$ $\lvert X - Y\rvert \leq \aleph_0 \wedge \lvert Y\rvert > \aleph_0$

Table 2.9: Nonconservative quantifiers.

Quantifier (Q)	Meaning of *Q Xes are Y* ($Q(X, Y)$)
oɸb⟩	$Y \subseteq X$
ɥɑoɸ	$Y \not\subseteq X$
ɸ⟩jh	$\lvert X\rvert = \lvert Y\rvert$
djɑɲ	$\lvert X\rvert < \lvert Y\rvert$
sjɒl	$\lvert X \cap Y\rvert \geq \lvert Y - X\rvert$

- If the next word is a quantifier, print that word.
- Otherwise, if there are no eligible outstanding descriptors, then print the next non-descriptor word.
- Otherwise, print the one whose antecedent falls the earliest and remove it from the list of outstanding descriptors.

2.6 | Pro-forms

Synopsis: *Pro-forms are words that replace a statement tree, and there are multiple kinds.*

2.6.1 | Pro-forms of the zeroeth kind

Strictly speaking, these are not a separate class of words, but rather a set of classless concretes:

- ⟨⟩lɑ⟩ the speaker or writer
- ⟨ɒɲj⟩ the listener or reader

Usually, these would mean *I* and *you*, respectively, but that does not always have to be the case.

2.6.2 | Pro-forms of the first kind

A previously mentioned concrete may, instead of receiving the usual class / mutability suffix, be referred by its first two segments plus the suffix ⟨–ω⟩: ⟨φnlf⟩ → ⟨φnω⟩. These pro-forms can be descriptored as with an ordinary concrete.

2.6.3 | Pro-forms of the second kind

This category is the most general of pro-concretes. Pro-forms of the second kind combine:

- A backref number: how many words deep? 0 means the previous concrete said by the same speaker, 1 the concrete before that and so on.
- A relation: describes the relation to the item referred to:
  - self
  - adversary
  - friend
  - parent (child)
  - teacher (student), and so on

The backref number is a single digit (which means that pro-forms of the second kind can look at only the previous 32 concretes). The relation can be one of the following:

Table 2.10: Relations for pro-forms of the second kind.

Root	Forward	Reverse
sfol	self	
ysnᵠω	adversary	
ndhy	friend, ally	
φjωl	parent	child
obll	teacher	student

If the forward relation is desired, then the backref number *follows* the relation. If the reverse relation (if applicable) is desired, then the backref number *precedes* the relation.

For instance, ⟨obllωd⟩ refers to a teacher of the previously-referred concrete, and ⟨ωdobll⟩ refers to a student of the previously-referred concrete.

2.6.4 | Pro-forms of the third kind

Unlike the other forms of pro-forms, this category deals with abstracts. There are only two:

- ⟨s⟩ refers to a temporal abstract (not necessarily the one that was mentioned the latest)
- ⟨no⟩ refers to a non-temporal abstract

The exact definition of a "temporal abstract" varies from person to person, but it is almost universally understood to include all of ⟨ɟɸɒ ɲnᵠ ⱺl ɒlˀ ɸɷh⟩.

Because pro-forms of the third kind are so short, they are seldom used with biting affixes attached. If biting affixes are needed, the abstract is almost always spelt in full.

2.6.5 | Pro-forms of the fourth kind

The phrase ⟨holˀ ɥɸo⟩ is an abstract that refers to a sentence previously said by the listener, and it is usually used alone. It can also modify a sentence that disambiguates what was referred to.

This phrase is commonly used either to show agreement or attach a connector to a sentence that previously did not admit any.

Note that this pro-form is composed of two words. A descriptor or connector can fall between the individual words of the phrase.

2.7 | Seasonal cycles

Some words are part of a quintuplet $(X_1, X_2, X_3, X_4, X_5)$ such that their meanings are rotated depending on the current season. They can represent concrete or abstract words, or even a mix of both.

Table 2.11: Seasonal cycling visualised for $(X_1, X_2, X_3, X_4, X_5)$ meaning $(m_1, m_2, m_3, m_4, m_5)$ in season 1.

Season	X_1	X_2	X_3	X_4	X_5
1	m_1	m_2	m_3	m_4	m_5
2	m_2	m_3	m_4	m_5	m_1
3	m_3	m_4	m_5	m_1	m_2
4	m_4	m_5	m_1	m_2	m_3
5	m_5	m_1	m_2	m_3	m_4

Perhaps the most well-known quintuplet is ⟨(ʃɟɸ, ɷsɟ, ɒɷnᵠ, hlɥ, olⱺ)⟩ which, in season 1, correspond to the five seasons in order. Thus, to refer to the second season, one would say:

- ⟨ɷsɟ⟩ during the first season
- ⟨hlɥ⟩ during the fourth season

A cycled concrete with its meaning currently m_i of some quintuplet of meanings $(m_1, \cdots, m_5)$ will have a class of *i*.

2.8 | Questions

In aaaaaaaaaA, questions are asked by placing an interrogative marker where the answer is desired and starting the sentence with the corresponding interrogative particle:

This replaces the top-level ▶, so there is no topic-fronting.

Note that interrogative markers are grammatically abstracts, even when a concrete answer is desired.

Table 2.12: Interrogative words in aaaaaaaaaA.

Particle	Marker	Definition
hʀɲ	ɾjh	what, who
ɾob	obɾ	when, where
slɴ	lns	why, how
ɸɥɴ	ɥnɸ	how many, how much
lhɑ	hɑl	to what extent

Responses are usually given using the particle followed by the answer:

\hʀɲ bo/ʀllbᴅ ɾjh ᴅɑ\ɾᴅɥᴅ
hʀɲ boɸ–>ɟ:lbᴅ ɾjh ᴅɑl–>l:ᴅɥᴅ
P-what eat-2-Ω Q-what bread-3-Ξ
Who ate the bread?

\hʀɲ ɭl/ᴅlbᴅ
hʀɲ ɭlɑ–>h:lbᴅ
P-what speaker-0-Ω
Me.

However, ⟨slɴ–lns⟩ questions are usually answered with full sentences.

⟨lhɑ⟩ and ⟨hɑl⟩ are special – they request a number, usually between 0 and 1, inclusive[1], that states to what degree the hypothesis is true. In that sense, they can be used to ask polar questions.

2.9 | Conjunctions

Synopsis: *A conjunction take the two nodes around it and returns a single node.*

Some common conjunctions are:

- ⟨ns⟩ *and* (for items), *or* (for predicates)
- ⟨slɟ⟩ *or* (for items), *and* (for predicates)
- ⟨nsnɟ⟩ *xor* (for both items and predicates)

Sometimes, it might be necessary to attach a node or descriptor to the conjunction and both of its arguments. In order to attach a node, the ⌜ (⌝) is attached to the end of a conjunction instead of the second (first) argument. Similarly, such descriptors are said to modify the conjunction itself.

\ɸ ᴅhɸᴅ ʀlɾ/ɸᴅlɥ \ɾɸ/ɸbhl n\ᴅbh ʀᴅ/ɥᴅlɥ
ɸ ᴅhɸᴅ ʀlɾᴅ–>n:ᴅlɥ >ɾɸl–>nᵠ:bhl ns–>ɾl:bh ʀᴅh–>n:ᴅlɥ
⌞ below mountain-5-Ξ⃗ DESC-sun-1-X⃗ and-⌝ valley-5-Ξ⃗

hnɾ lᴅfh n\lfn hnᵠɾ bɾ/ʀhsl ɭhɥ
hnɾ lᴅfh ns–>ɥ:fn hnᵠɾ bɾᴅ–>h:ɑsl ɭhɥ
create hope and-⌜ life person-0-Φ all

[1] A value less than 0 or greater than 1 can be interpreted as an emphatic answer.

The mountains and valleys below the sun creates hope and life for all.

2.9.1 | Appositives

The conjunction ⟨ɒ⟩ = joins the two nodes around it and returns a single node that equates its arguments and refers to the entity in question.

\ll/jɒʏɒ jɑɒ ʏɒ\jlb/lʃn ɑj\ɑʃn ʃnᵠj b /ʃɑˈlɑ b ɒ
lɑ–>h:ɒʏɒ jɑɒ ʏɒɑ–>l:lbɒ–>ʏ:ʃn ɑjɸ–>ʏ:ʃn ʃnᵠj b >ʃɑˈlɑ b ɒ
speaker-0-Ξ walk village-3-Ω-⌜ association-⌜ uncle ⌟ assocation.DESC-speaker ⌟ =

ʏɒ\jbh/ɒʃn bnɒb h \ɒhɸ
ʏɒɑ–>l:bhl–>ʏ:ʃn bnɒb h >ɒhɸ
village-3-$\vec{X}$-⌜ large DUMMY DESC-superlative

We visited my uncle's village, the largest village in the world.

2.10 | Subordinate clauses

Subordinate clauses are conceptually questions embedded as nodes. These are content clauses by default, but combined with appositives, they can act as relative clauses. In subordinate clauses, the airflow direction is changed immediately before the question word and the clause is closed with the particle ⟨b⟩ (if there is anything afterward):

\ll/jɒʏɒ jɑɒ ɒhɾlo \ʃob nɑ lɑ ɾlb/ɾɒʏɒ lh h \obʃ
lɑ–>h:ɒʏɒ jɑɒ ɒhɾlo >ʃob nɑ lɑ ɾlbɑ–>nᵠ:ɒʏɒ lh h >obʃ
speaker-0-Ξ walk again SUB-where see speaker rdkbe$_5$-1-Ξ existence DUMMY DESC-Q-where

We went back to the place where we saw the roses. (said in second season)

\lɾlɒ/nɒʏɒ bll ɒ \hɾlj lh ɒnɑ ɒɑʃonᵠ /ɑsn \ll'ɾjh b
lɾlɒl–>nᵠ:ɒʏɒ bll ɒ >hɾlj lh ɒnɑ ɒɑʃonᵠ >ɑsn >ll'ɾjh b
young_person-1-Ξ finite = SUB-who existence wreath head DESC-flower borrow.DESC-Q.who ⌟

ɒlbjʃ lljl h /bhʏ
ɒlbjʃ lljl h >bhʏ
dance environs DUMMY DESC-fire

Many little girls with wreaths of flowers on their heads danced around the bonfire.

2.11 | Connectors

Synopsis: *Connectors are free-floating particles that establish relationships between sentences.*

Connectors are special particles that can be placed anywhere in the sentence *(other than at the beginning)* and exist independently from the rest of the sentence. In other words:

- A connector alone cannot separate a descriptor from its antecedent.
- A connector can separate a quantifier from its antecedent, as long as no non-connectors separate the two.

Connectors a and b are part of the same *set* S iff all of the following conditions hold:

- a and b are identical (and are of the same parity)
- they belong to sentences α and β, respectively (NB: it is possible that $\alpha = \beta$)
- if a is the ith word of α, then b is the ith word of β
- there are no sentences between α and β whose ith word is a connector different from a and b (or is of different parity)

For the purposes of positioning connectors, two consecutive instances of ⟨b⟩ ⌟ within the same sentence is treated as one word.

Note that "belonging to the same connector set" is an equivalence relation.

Table 2.13: Connectors.

Name	Arity	aaaaaaaaaaA	Explanation
Sequential	*n*	ɥʃ	A sequence of events. An event α is said to happen before β if the sentence describing α is uttered before that describing β.
Parallel	*n*	ʏsɭj	Two or more events happening simultaneously.
Analogous	2	ʏhɷ	"For the same reason α is true, β is also true." Also used as an "and" without stating any order.
Subversive	*n*	ʏlo	"θ_1 but θ_2 but θ_3 but etc."
Augmentative	*n*	ʏɸb	Later statements apply to a greater extent than earlier statements.
Explanatory	*n*	ʏls	"θ_1 causes θ_2 causes θ_3 etc."

Sentences of a connector set are joined by the relation of the connector used therein:

\ɸnɭ/ɹɒʏɒ boɸ ɥʃ ɷsn.
ɸnɭʃ–>ɻ:ɒʏɒ boɸ ɥʃ ɷsn
fish-2-Ξ eat SEQUENTIAL flower
The fish ate the flower.

\lɾɒ/ɸlbɒ ɒlbɻʃ ɥʃ ɭlɻl \ɒol.
lɾɒl–>nᵠ:lbɒ ɒlbɻʃ ɥʃ ɭlɻl >ɒol
young_person-1-Ω dance SEQUENTIAL surroundings DESC-tree
Then the child danced around the tree.

\lɾɷ boɸ ɥʃ ɸnɷ.

ʅɦΩ boφ ɥʃ φnΩ
young_person#PF1 eat SEQUENTIAL fish#PF1
Then the child ate the fish.

The polarity of a connector can be flipped by flipping the least significant bit of the rightmost segment.

3 | Numerals

Synopsis: *aaaaaaaaaA supports a variety of bases, and even mixed bases as with Lek-Tsaro numerals up to 4199. However, nonrectangular number systems are unsupported.*

The basic digits are as follow:

Table 3.1: Basic digits.

#	#	Word	#	#	Word
ᚻ	0	ɑɒ	ᚽ	16	sʝ
ᚾ	1	sʃ	ᚿ	17	ɾʝ
ᚿ	2	snᵠ	ᛀ	18	sn
ᛁ	3	sɥ	ᛁ	19	ɾn
ᛂ	4	sb	ᛂ	20	ɾl
ᛃ	5	ɾl	ᛃ	21	ɾnᵠ
ᛃ	6	sl	ᛃ	22	ɑl
ᛄ	7	ɾo	ᛄ	23	ɑɥ
ᛇ	8	sɒ	ᛇ	24	sɸ
ᛈ	9	sɭ	ᛈ	25	ɾh
ᛊ	10	ɾɸ	ᛊ	26	ɑh
ᛋ	11	ɑɸ	ᛋ	27	sh
ᛍ	12	so	ᛍ	28	ɑn
ᛎ	13	ɑo	ᛎ	29	ɑh
ᛏ	14	ɾl	ᛏ	30	ɾb
ᛐ	15	ɾʃ	ᛐ	31	ɑʃ

Note that if a segment starts the name of any digit, it cannot end the name of any digit, and vice versa, allowing for chaining without triggering phonotactic violations.

Thus a numeral is parsed as such:

- Set $r := 1, n := 0, c := 1$.
- For each two-segment chunk, let a be the digit represented by the segments. Then:
  - If airflow is opposite of starting airflow, then:
    - If this is the first chunk, then reject.
    - Otherwise, set $r := 32$ if $a = 0$, else $r := a$.
  - Otherwise, if $r = 1$ and this is not the first chunk, reject.

  - Otherwise, set $c := r \cdot c$ then $n := n + a \cdot c$.

- Return n.

For instance, ⟨\ɾdl/sɹ\syso⟩ represents $C35_{16} = 3125_{10}$. This number can also be written:

\ɷᴅ/ɾh\ɷᴅɾl
[illegible]
50P0

Note that the numeral is spelled out in little-endian order, but displayed using figures in big-endian.

An instance of ⟨ɾᴅ⟩ followed by a digit d is equivalent to a series of d zeroes, so the same number can be written ⟨\ɷᴅ/ɾl\ɾᴅsbsɾ⟩.

Gramatically, cardinal numerals are considered quantifiers. Ordinal numerals look the same, but are considered descriptors and bear an airflow switch at their beginning.

3.1 | Fractional numerals

Fractional parts of a numeral are set off after the integral part (if present), separated by a fractional point ⟨◇⟩, read ⟨ɷɹ⟩. Then the fractional part is parsed as follows:

- Set $r := 1, n := 0, c := 1$.
- For each two-segment chunk, let a be the digit represented by the segments. Then:
  - If airflow is opposite of starting airflow, then set $r := 32$ if $a = 0$, else $r := a$. Then set $c := r$.
  - Otherwise, if $r = 1$, reject.
  - Otherwise, set $c := r \cdot c$ then $n := n + a/c$.
- Return n.

For instance, $2 \cdot \pi$ can be approximated as $6.487ED5_{16}$, which is read:

\slɷɹ/sɹ\sbsᴅɾoɾlɷoɾl
[illegible]
6.G487ED5

In both the spelled-out version and the version using the figures, the digits after the decimal point are in big-endian order.

3.2 | Figures in signage

In signage, single-digit integers are not displayed alone due to possible confusion with digits in other writing systems. Instead, a filler ".0" is added, so, for instance, ⟨[illegible]⟩ is displayed as ⟨[illegible]⟩. This happens even in contexts in which fractional numbers do not make sense.

If other languages are used on a sign alongside aaaaaaaaaA (i. e. seldom within Nŋln), then these languages receive similar treatment.

4 | Word formation

4.1 | Compounding

Synopsis: *Compounding in aaaaaaaaaA is done by interleaving the segments between the base words.*

aaaaaaaaaA distinguishes between *coördinating* and *subordinating* compounds. Coördinating compounds place both of their constituents at the same level – for instance, in *flint and steel*, *flint* and *steel* are represented equally. Subordinating compounds place one of their constituents as a dependent of the other – an *elderberry* is a type of berry, not a type of elder. Quite naturally, subordinating compounds are more common than coördinating compounds.

A list of words is said to be *trivially interleavable* if one of the following holds:

- All words have an equal number of segments (eq).
- All words but one have an equal number of segments, and the one remaining has one more segment than the others (aug).
- All words but one have an equal number of segments, and the one remaining has one fewer segment than the others (dim).

If a list of words S is not trivially interleavable, then the following steps are taken:

- Initialise S' to S.
- While S' is not trivially interleavable:
  - Find the shortest word in S' and its index i.
  - Append a copy of $S[i]$ to $S'[i]$.

Obviously an (eq)-interleavable list of words can be interleaved in that order, so ⟨φlh, ɑsn, ʃφl⟩ can be interleaved into ⟨φɑʃlsφhnl⟩.

In an (aug)-interleavable list of words, the longest word must be interleaved first, as to have its last segment end the compound. In order to disambiguate the order of the consitutents, if the longest word is not also the first, then its index is prefixed: ⟨jɑɒ, φlh, φlɾdɒ⟩ makes ⟨snᵠφjφlɑlɾdɒhɒ⟩.

Similarly, in a (dim)-interleavable list, the shortest word must be interleaved last. If the shortest word is not also the last, then its index *from the end* is prefixed: ⟨ɒhɾdo, ɒhφ, bnɒb⟩ makes ⟨sʃɒbɒhnhɾdφob⟩.

4.1.1 | Coördinating compounds

It is natural that coördinating compounds can involve any number of constituents, which can usually be reordered at will. A compound from a (eq)-interleavable list of words receives no marking; other coördinating compounds receive the ⟨–φn⟩ suffix.

4.1.2 | Subordinating compounds

Unlike in cöordinating compounds, the constituents of a subordinating compound is order-dependent. In particular, the constituents are put in head-initial order with right-associativity. For instance, ⟨φlh, ɑsn, ſφl⟩ *morning*, *flower*, *sun* means *morning of sunflowers* (not *flower-morning of the sun*).

A subordinating compound from a (eq)-interleavable list of words receives the ⟨–φn⟩ suffix; other compounds receive no marking.

A | Dictionary

An entry looks like this:
ɒll *a* loan (1) is borrowing (0) [from (2)] (modifying) (0)'s (*) (borrowed)
From left to right:

1. The entry – the aaaaaaaaaaA term listed.
2. The part of speech of the corresponding entry:
    - *c* – a concrete
        - *c1* – *c5* – of one of five season classes
        - *c0* – season-neutral concrete
    - *a* – an abstract
3. The definition – the gloss for the corresponding entry.
    (a) (0) – the first argument when used as a stem in an expression tree
    (b) (1) – the second argument, and so on
    (c) (*) – parent (antecedent) of expression tree
4. If applicable, any special grammatical or semantic notes for this term.
5. Optionally, examples of usage.

| (

(ʃjɸ, ɑsj, ɒɑnᵠ, hlɥ, olɾ) *qccccc* the five seasons (LSp-ESu, LSu-EAu, MAu-EWi, MWinter, LWi-ESp)

(ɒjɸ, jhɒ, slb, ɑnl, hnᵠʃ) *qcccaa* (asparagus, cucumber, carrots, perception, life)

(jsb, ɥlɥ, jʃl, bɸl, ɾbɑ) *qccccc* (rose, dandelion, kosmos, beetroot, elderberry)

(ɸlɑ, blj, ɑlh, nᵠoɑ, snɒ) *qccccc* (strawberry, peach, pumpkin, icefruit, citrus fruit) The icefruit is an unusual fruit that grows partially underground during the winter. It has a moderately thin skin and its flesh has the consistency of apples, but tastes slightly bitter when raw.

| ʃ

ʃnᵠj *c0* uncle, aunt
ʃbɑ *a* finish (modifying) terminative, perfect

ʃɸl *c1* sun

b

bʃɒ *c0* person, entity
boɸ *c2* food (0) eats (1)
bnɒb *a* size, large
bhɥ *c3* fire

ɒ

ɒlˀ *a* start (modifying) inceptive, inchoative
ɒoˀ *c2* tree
ɒɸjs *c0* weather
ɒlbjʃ *c2* dance
ɒˀl *a* loan (1) is borrowing (0) [from (2)] (modifying) (0)'s (*) (borrowed)
ɒnɷ *c1* wreath
ɒɾj *c0* the listener or reader, you
ɒɷʃonᵠ *c5* head
ɒɷl *c3* bread
ɒhɸ *a* edge, boundary, extremity, superlative (of (0))
ɒhɸɒ *a* below, underside (modifying) (*) is under, below (0); downward
ɒhɾlo *a* again, repeat

j

jnᵠ *a* after
jɸɒ *a* time when (0), (1) (modifying) (*) happened at the time (0)
jɷɒ *c1* walk (toward (1))

ɥ

ɥɒɷ *c3* village, town

ɸ

ɸlɾdɒ *c4* dark, dim
ɸlh *c0* morning
ɸnˀlʃ *c2* fish
ɸɷh *a* absence it is not true that (0)

l

lɒʃh *a* hope

ˀ

ˀblɷ *c1* summer solstice
ˀljˀ *a* environs, (0) surrounds (1) (modifying) (*) surrounds (0), around (0)
ˀlɷ *c0* the speaker or writer, I
ˀɾdɒˀ *c1* young person
ˀh *a* existence, (0) is at (1) (modifying) (*) is at (0)

n

nɥ *a* destination (0) intends to go to (1) (modifying) to (0)
nɷ *a* sight (0) sees (1)

ɾ

ɾlʃɒ *c5* mountain
ɾdh *c5* valley, crease
ɾl *a* before

ɷ

ɷjɸ *a* association, relationship (0) is related to (1) [by (2)], (0) is (1)'s (2) (by association) (modifying) (0)'s (*) (associated)
ɷsn *c1* flower

h

hɸb *a* property, belongings (1) is the owner of (0) (modifying) (0)'s (*) (owned)
hnʃ *a* creation (0) creates (1) (modifying) (*) created by (0)