aaaaaaaaaaA, the language of *somewhere*

uruwi

aaaaaaaaaaaaaaaaa

A complete grammar

13 December 2017

Dedicated to Mareck.

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

(C)opyright 2017 Uruwi. See README.md for details.

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	ſ	t’	!	e˥
1	n$^{\varphi}$	ŋ	ɠ	ã
2	b	p	p	y
3	ᴅ	m	ɓ	ũ
4	J	s	s	ɨ
5	ɥ	j	j	i
6	ɭ	tɬ’	‖	ʌ˥
7	o	w	w	u
8	φ	k	k	o
9	l	l	l	ɯ
10	ɿ	k’	ǂ	o˥
11	n	n	n	ẽ
12	ɗ	r	ɹ	ḛ
13	Ω	t	t	e
14	h	ħ	h	a
15	s	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:

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

(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 ⟨–>ɥ:ʃn⟩ on ⟨ᴅΩl⟩, then we would:

- Take the last letter of ⟨ᴅΩl⟩, namely ⟨–l⟩ (9), and xor it with ⟨–ɥ⟩ (5). $9 \veebar 5 = 12$ so we now have ⟨–ɦ⟩.
- Append ⟨–ʃn⟩. We now have ⟨ᴅΩ>ɦʃn⟩.

2 | The statement space

2.1 | Conceptualisation

Synopsis: *aaaaaaaaaaA 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

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

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

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

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 aaaaaaaaaaA assigns to the special symbols:

Table 2.2: Names of syntactic markers in aaaaaaaaaaA.

△	ȷs
▶	h
⌜	−>ɥ:ʃn
⌟	b
⌞	φ
⌝	−>ɗ:bh

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 aaaaaaaaaaA.

#	(° 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	–>ȷ:
3	–>l:
4	–>b:
5	–>n:
0	–>h:

Sym	Affix 2
Ω	lbᴅ
Ψ	lɷnᵠ
X	bɷȷ
Z	ɭʃɥ
Ξ	ᴅɥᴅ
Φ	ɷsꞁ
$\vec{\Omega}$	lᴅɥ
$\vec{\Psi}$	lhᴅ
$\vec{X}$	bhɭ
$\vec{\Xi}$	ᴅɭɥ
$\vec{\Phi}$	ɷʃꟼ
Σ	ɭnᵠo

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:

- ⟨boφ⟩ *c2* (0) eats (1)
- ⟨φnꞁʃ⟩ *c2* fish
- ⟨ɷsn⟩ *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:

\φnꞁ/ȷᴅɥᴅ bo\ɦᴅɥᴅ ɷs/ꞁᴅɥᴅ
φnꞁʃ–>ȷ:ᴅɥᴅ boφ–>ȷ:ᴅɥᴅ ɷsn–>nᵠ:ᴅɥᴅ
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:

\φnꞁ/ȷᴅɥᴅ boφ ɷsn
φnꞁʃ–>ȷ:ᴅɥᴅ boφ ɷsn
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 ᴅφȷ\nᵠɑs/ɩbh φ ٦bφ\٦bɑȷ ȷ/ɑbh φl\nᵠᴅh
φ φ ſφl–>nᵠ:bɑȷ ᴅφȷs–>h:ɑs٦–>ɗ:bh φ ٦blɑ–>ȷ:bɑȷ ȷnᵠ–>ɗ:bh φlh–>ɗ:bh
⌞ ⌞ sun-1-$\vec{X}$ weather-0-Φ-⌝ ⌞ summer_solstice-2-$\vec{X}$ after-⌝ morning($\vec{X}$)-⌝

The rest of the sentence is thus:

ȷφᴅ ȷɑ/bᴅy\ɩſn ٦lɑ ns/nᵠn ɗ/\φᴅly b ᴅl٦
ȷφᴅ ȷɑᴅ–>nᵠ:ᴅyᴅ–>y:ſn ٦lɑ ny–>y:ſn ɗſᴅ–>n:ᴅly b ᴅl٦
time walk-1-Ξ-⌜ speaker(Ξ) destination-⌜ mountain-5-$\vec{\Xi}$ ⌟ 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:

⟨φlh⟩ Morning → ⟨>φlh⟩ D(Morning)
⟨ȷφᴅ⟩ Time → ⟨>ȷφᴅ⟩ D(Time)

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

\φ ᴅφȷ/nᵠΩsl ȷnᵠ \ſφl /\blΩ φl\nᵠᴅh
φ ᴅφȷs–>h:Ωsl ȷnᵠ >ſφl >lblΩ φlh–>ɗ:bh
⌞ weather-0-Φ after D-sun D-summer_solstice morning-⌝

ȷφᴅ ȷΩ/bᴅɥ\lſn nɥ /\lΩ \ɗlᴅ /ᴅl)
ȷφᴅ ȷΩᴅ–>nᵠ:ᴅɥᴅ–>ɥ:ſn nɥ >llΩ >ɗlᴅ >ᴅl)
time walk-1-Ξ-⌜ destination D-speaker D-mountain D-start

Other descriptors are possible:

Table 2.7: Other descriptors.

Prefix	Input	Output
>ſ'	C	Inalienable_Possession(C)
>ſΩ'	C	Association(C)
>φb'	C	Property(C)
>ll'	C	Borrow(C)
>ȷh'	C	Destination(C)

2.5.2 | Quantifiers

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

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

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 ($Q(X, Y)$)
ɿhɥ	All	$X \subseteq Y$
ɥnᵠɿ	Not all	$X \not\subseteq Y$
Ωɥφ	Some	$(X \cap Y) \neq \varnothing$
bɿo	None	$(X \cap Y) = \varnothing$
ᴅɥo	Most	$\lvert X \cap Y\rvert \geq \lvert X - Y\rvert$
φɿh	At least two	$\lvert X \cap Y\rvert \geq 2$
onφ	One	$\lvert X \cap Y\rvert = 1$
ᴅnnᵠ	Half of	$\lvert\lvert X \cap Y\rvert - \lvert X - Y\rvert\rvert \leq 1$
bɭl	A finite number of (sometimes “many”)	$\lvert X \cap Y\rvert < \aleph_0$
sȷh	A countable number of	$\lvert X \cap Y\rvert \leq \aleph_0$
Ωlɭ	An infinite number of	$\lvert X \cap Y\rvert \geq \aleph_0$
ſnnᵠ	An uncountable number of	$\lvert X \cap Y\rvert > \aleph_0$
lȷl	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 ($Q(X, Y)$)
oφbɿ	$Y \subseteq X$
ɥΩoφ	$Y \not\subseteq X$
φlȷh	$\lvert X\rvert = \lvert Y\rvert$
dȷΩᶇ	$\lvert X\rvert < \lvert Y\rvert$
sȷᴅɭ	$\lvert X \cap Y\rvert \geq \lvert Y - X\rvert$

- 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:

- ⟨ɿlΩ⟩ the speaker or writer
- ⟨ᴅᶇȷ⟩ 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 ⟨–ꭥ⟩: ⟨φn⌉⌈⟩ → ⟨φnꭥ⟩. 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
s⌈o⌉	self	
ɥsnᵠꭥ	adversary	
ɗᴅhɥ	friend, ally	
φȷꭥ⌉	parent	child
obl⎩	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, ⟨obl⎩ꭥᴅ⟩ refers to a teacher of the previously-referred concrete, and ⟨ꭥᴅobl⎩⟩ 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 ⟨ȷφᴅ ȷnᵠ ɲl ᴅɭɿ φΩh⟩.

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.

2.7 | 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 ⟨(ɾȷφ, ΩSȷ, ᴅΩnᵠ, hɿɥ, olɲ)⟩ which, in season 1, correspond to the five seasons in order. Thus, to refer to the second season, one would say:

- ⟨ΩSȷ⟩ during the first season
- ⟨hɿɥ⟩ 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 aaaaaaaaaaA, questions are asked by placing an interrogative marker where the answer is desired and starting the sentence with the corresponding interrogative particle:

Table 2.12: Interrogative words in aaaaaaaaaaA.

Particle	Marker	Definition
hɲȷ	ɲȷh	what, who
ɾob	obɾ	when, where
sln	lns	why, how
φɥn	ɥnφ	how many, how much
ɭhΩ	hΩɭ	to what extent

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.

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

\hɾ|j bo/ɾlbᴅ ɾ|jh ᴅɑ\ſᴅцᴅ
hɾ|j boφ–›j:lbᴅ ɾ|jh ᴅɑl–›l:ᴅцᴅ
P-what eat-2-Ω Q-what bread-3-Ξ
Who ate the bread?

\hɾ|j ˥l/ᴅlbᴅ
hɾ|j ˥lɑ–›h:lbᴅ
P-what speaker-0-Ω
Me.

However, ⟨sln–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 | Appositives

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

\˥l/jᴅцᴅ jɑᴅ цᴅ\jlb/lſn ɑj\ɑſn ſnᵠj b /ſɑ'˥lɑ b ᴅ
˥lɑ–›h:ᴅцᴅ jɑᴅ цᴅɑ–›l:lbᴅ–›ц:ſn ɑjφ–›ц:ſn ſnᵠj b ›ſɑ'˥lɑ b ᴅ
speaker-0-Ξ walk village-3-Ω-⌜ association-⌜ uncle ⌟ assocation.DESC-speaker ⌟ =

цᴅ\jbh/ᴅſn bnᴅb h \ᴅhφ
цᴅɑ–›l:bhl–›ц:ſn bnᴅb h ›ᴅhφ
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):

\˥l/jᴅцᴅ jɑᴅ ᴅhɾo \ſob nɑ ˥lɑ ɾb/ɾᴅцᴅ ˥h h \obſ
˥lɑ–›h:ᴅцᴅ jɑᴅ ᴅhɾo ›ſob nɑ ˥lɑ ɾbɑ–›nᵠ:ᴅцᴅ ˥h h ›obſ
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)

\˥ɾᴅ/nᴅцᴅ bl| ᴅ \hɾ|j ˥h ᴅnɑ ᴅɑſonᵠ /ɑsn \˥l'ɾ|jh b

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

ໄძᴅໄ–›ռᵠ:ᴅɣᴅ bꞁꞁ ᴅ ›hძȷ ໄh ᴅռΩ ᴅΩʃoռᵠ ›Ωsռ ›ໄꞁ'ძȷh b
young_person-1-Ξ finite = SUB-who existence wreath head DESC-flower borrow.DESC-Q.who ⌟

ᴅlbȷʃ ໄꞁȷໄ h /bhɥ
ᴅlbȷʃ ໄꞁȷໄ h ›bhɥ
dance environs DUMMY DESC-fire

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

3 | Numerals

Synopsis: *aaaaaaaaaaA 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
⊦	0	ΩD	⊦̦	16	sȷ
ᒪ	1	sʃ	ᒪ̦	17	dȷ
ƽ	2	sn$^{\varphi}$	ƽ̦	18	sn
ϟ	3	sɥ	ϟ̦	19	dn
ʔ	4	sb	ʔ̦	20	dl
≀	5	dl	≀̦	21	dn$^{\varphi}$
ↄ	6	sl	ↄ̦	22	Ωl
⌉	7	do	⌉̦	23	Ωɥ
⌈	8	sD	⌈̦	24	sφ
ᑕ	9	s⌉	ᑕ̦	25	dh
Ƨ	10	dφ	Ƨ̦	26	Ωh
ς	11	Ωφ	ς̦	27	sh
ɾ	12	so	ɾ̦	28	Ωn
ɛ	13	Ωo	ɛ̦	29	Ωh
ᒧ	14	dl	ᒧ̦	30	db
⊣	15	dʃ	⊣̦	31	Ωʃ

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, ⟨\ɾl/sȷ\sɥso⟩ represents $C35_{16} = 3125_{10}$. This number can also be written:

\ɑᴅ/ɾh\ɑᴅɾl
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 ⟨\ɑᴅ/ɾl\ɾᴅsbsʃ⟩.

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 ⟨ɑȷ⟩. 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 $r := a$.
  - 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:

\slɑȷ/sȷ\sbsᴅɾoɾlɑoɾl
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, ⟨ᓴ⟩ is displayed as ⟨ᓴ◦ᒻ⟩. This happens even in contexts in which fractional numbers do not make sense.

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

4 | Word formation

4.1 | Compounding

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

aaaaaaaaaaA 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, ʊsn, ɾφl⟩ can be interleaved into ⟨φʊɾlsφhnl⟩.

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: ⟨ȷʊᴅ, φlh, φlɦᴅ⟩ makes ⟨sn^{φ}φȷφlʊlɦᴅhᴅ⟩.

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: ⟨ᴅhɦo, ᴅhφ bnᴅb⟩ makes ⟨sɾᴅbᴅhnhɦᴅφob⟩.

4.1.1 | Coördinating compounds

It is natural that coördinating compounds can involve any number of constituents

A Dictionary

An entry looks like this:

ᴅľl *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.

(

(ʃȷφ, ɑsȷ, ᴅɑnᵠ, hľɥ, oľd) *qccccc* the five seasons (LSp-ESu, LSu-EAu, MAu-EWi, MWinter, LWi-ESp)

(ᴅȷφ, ȷhᴅ, sľb, ɑnl, hnᵠʃ) *qcccaa* (asparagus, cucumber, carrots, perception, life)

(ȷsb, ɥľɥ, ȷʃl, bφl, ɗbɑ) *qccccc* (rose, dandelion, kosmos, beetroot, elderberry)

(φľɑ, blȷ, ɑlh, nᵠoɑ, snᴅ) *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

ʃnᵠȷ *c0* uncle, aunt

ʃbɑ *a* finish (modifying) terminative, perfect ɾφl *c1* sun

b

boφ *c2* food (0) eats (1)
bnᴅb *a* size, large
bhɥ *c3* fire

ᴅ

ᴅlɿ *a* start (modifying) inceptive, inchoative
ᴅφȷs *c0* weather
ᴅlbȷɾ *c2* dance
ᴅɿl *a* loan (1) is borrowing (0) [from (2)] (modifying) (0)'s (*) (borrowed)
ᴅnΩ *c1* wreath
ᴅɲȷ *c0* the listener or reader, you
ᴅΩɾonᵠ *c5* head
ᴅΩl *c3* bread
ᴅhφ *a* edge, boundary, extremity, superlative (of (0))
ᴅhɲo *a* again, repeat

ȷ

ȷnᵠ *a* after
ȷφᴅ *a* time when (0), (1) (modifying) (*) happened at the time (0)
ȷΩᴅ *c1* walk (toward (1))

ɥ

ɥᴅΩ *c3* village, town

φ

φlɲᴅ *c4* dark, dim
φlh *c0* morning
φnɿɾ *c2* fish
φΩh *a* absence it is not true that (0)

ɿ

ɿblΩ *c1* summer solstice
ɿlȷɿ *a* environs, (0) surrounds (1) (modifying) (*) surrounds (0), around (0)
ɿlΩ *c0* the speaker or writer, I
ɿɲᴅɿ *c1* young person
ɿh *a* existence, (0) is at (1) (modifying) (*) is at (0)

n

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

ɲ

ɲɾᴅ *c5* mountain
ɲl *a* before

Ω

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

h

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