aaaaaaaaaaA, the language of *somewhere*

uruwi

aaaaaaaaaaaaaaaaa

A complete grammar

16 November 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	ʃ	t’	!	e˥
1	nᵠ	ŋ	ɠ	ã
2	b	p	p	y
3	ᴅ	m	ɓ	ũ
4	ɟ	s	s	ɨ
5	ɥ	j	j	i
6	ɭ	tɬ’	ǁ	ʌ˥
7	o	w	w	u
8	φ	k	k	o
9	l	l	l	ɯ
10	ʅ	k’	ǂ	o˥
11	n	n	n	e˥
12	ɗ	r	ɹ	ḛ
13	ʊ	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˩↓].

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 modifier
- at certain affixes
- in the middle of certain roots

On a proper noun, as well as on encountering a nasal vowel, 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:

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

(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 ⟨–>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 ⟨–nl⟩.
- Append ⟨–ʃn⟩. We now have ⟨ᴅΩ>nlʃn⟩.

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

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

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

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

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 aaaaaaaaaaA assigns to the special symbols:

Table 2.2: Names of syntactic markers in aaaaaaaaaaA.

△	ʝs
▶	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 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
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.

Symbol	Name	Description
$\vec{\Psi}$	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	–>n$^{\varphi}$:
2	–>ȷ:
3	–>l:
4	–>b:
5	–>n:
0	–>h:

Sym	Affix 2
Ω	lbᴅ
Ψ	lΩn$^{\varphi}$
X	bΩȷ
Z	ʅʃɥ
Ξ	ᴅɥᴅ
Φ	Ωsɿ
$\vec{\Omega}$	lᴅɥ
$\vec{\Psi}$	lhᴅ
$\vec{X}$	bhʅ
$\vec{\Xi}$	ᴅʅɥ
$\vec{\Phi}$	Ωʃɿ
Σ	ʅn$^{\varphi}$o

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)
- ⟨φnɿʃ⟩ *c2* fish

- ⟨Ωsn⟩ *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:

\φnɿ/jᴅɥᴅ bo\ɗᴅɥᴅ Ωs/ɿᴅɥᴅ
φnɿſ–>j:ᴅɥᴅ boφ–>j:ᴅɥᴅ Ωsn–>n^{φ}: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:

\φnɿ/jᴅɥᴅ boφ Ωsn
φnɿſ–>j:ᴅɥᴅ boφ Ωsn
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/lbΩj ᴅφj\n^{φ}Ωs/ʅbh φ ɿbφ\ɿbΩj j/Ωbh φl\n^{φ}ᴅh
φ φ ſφl–>n^{φ}:bΩj ᴅφjs–>h:Ωsɿ–>ɗ:bh φ ɿblΩ–>j:bΩj jn^{φ}–>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:

jφᴅ jΩ/bᴅɥ\ʅſn ɿlΩ ns/n^{φ}n ɗ/\φᴅlɥ b ᴅʅɿ
jφᴅ jΩᴅ–>n^{φ}:ᴅɥᴅ–>ɥ:ſn ɿlΩ nɥ–>ɥ:ſn ɗſᴅ–>n:ᴅlɥ b ᴅʅɿ
time walk-1-Ξ-⌜ speaker(Ξ) destination-⌜ mountain-5-$\vec{Ξ}$ ⌟ 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:ɑsꞁ ȷnᵠ ›ʃφl ›ꞁblɑ φlh–›ɗ:bh
⌞ weather-0-Φ after D-sun D-summer_solstice morning-⌝

ȷφᴅ ȷɑ/bᴅɥ\lʃn nɥ /\lɑ \ɗlᴅ /ᴅlꞁ
ȷφᴅ ȷɑᴅ–›nᵠ:ᴅɥᴅ–›ɥ:ʃn nɥ ›ꞁlɑ ›ɗlᴅ ›ᴅlꞁ
time walk-1-Ξ-⌜ destination D-speaker D-mountain D-start

Other descriptors are possible:

Table 2.7: Other descriptors.

Prefix	Input	Output
›ʃ'	*C*	Inalienable_Possession(*C*)
›ʃɑ'	*C*	Association(*C*)
›φb'	*C*	Property(*C*)
›ꞁl'	*C*	Borrow(*C*)

2.5.2 | Quantifiers

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

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

The following are some quantifiers in aaaaaaaaaaA:

Table 2.8: Conservative quantifiers.

Quantifier (Q)	Translation	Meaning of Q *Xes are Y* ($Q(X, Y)$)
ꞁhɥ	All	$X \subseteq Y$
ɥnᵠꞁ	Not all	$X \not\subseteq Y$
ɑɥφ	Some	$(X \cap Y) \neq \varnothing$
bꞁo	None	$(X \cap Y) = \varnothing$
ᴅɥo	Most	$\|X \cap Y\| \geq \|X - Y\|$
φlh	At least two	$\|X \cap Y\| \geq 2$
onφ	One	$\|X \cap Y\| = 1$
ᴅnnᵠ	Half of	$\|\|X \cap Y\| - \|X - Y\|\| \leq 1$
blꞁ	A finite number of	$\|X \cap Y\| < \aleph_0$
sȷh	A countable number of	$\|X \cap Y\| \leq \aleph_0$
ɑlꞁ	An infinite number of	$\|X \cap Y\| \geq \aleph_0$
ʃnnᵠ	An uncountable number of	$\|X \cap Y\| > \aleph_0$

All of the above quantifiers are *conservative* – in other words, $Q(X, Y) \iff Q(X, X \cap Y)$. aaaaaaaaaaA has determiners where this is not the case:

Table 2.9: Nonconservative quantifiers.

Quantifier (Q)	Meaning of Q *Xes are* Y ($Q(X, Y)$)
oφbl	$Y \subseteq X$
ɥɒoφ	$Y \not\subseteq X$
φlɟh	$\lvert X\rvert = \lvert Y\rvert$
dɟɒɼl	$\lvert X\rvert < \lvert Y\rvert$
sɟɒl	$\lvert X \cap Y\rvert \geq \lvert Y - X\rvert$

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:

- 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:

- ⟨ꞁlɒ⟩ the speaker or writer
- ⟨ɒɼɟ⟩ 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 ⟨–ɒ⟩: ⟨φnꞁɼ⟩ → ⟨φnɒ⟩. 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:

1. 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.

2. A relation: describes the relation to the item referred to.

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	ꭥᴅ	⊦	16	sȷ
⌊	1	sʃ	⌊	17	ɾȷ
ς	2	snᵠ	ς	18	sn
ʢ	3	sɥ	ʢ	19	ɾn
ʔ	4	sb	ʔ	20	ɾɳ
≀	5	ɾɭ	≀	21	ɾnᵠ
⊃	6	sɭ	⊃	22	ꭥl
⌉	7	ɾo	⌉	23	ꭥɥ
⌈	8	sᴅ	⌈	24	sφ
⊂	9	sʅ	⊂	25	ɾh
ʕ	10	ɾφ	ʕ	26	ꭥh
ʕ	11	ꭥφ	ʕ	27	sh
ɿ	12	so	ɿ	28	ꭥn
ɩ	13	ꭥo	ɩ	29	ꭥh
⌋	14	ɾl	⌋	30	ɾb
⊣	15	ɾʃ	⊣	31	ꭥʃ

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 $r := a$.
  - 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\syso⟩ represents $C35_{16} = 3125_{10}$. This number can also be written ⟨\ꭥᴅ/ɾh\ꭥᴅɾl⟩.

A | Dictionary

An entry looks like this:

ᴅ𝗅𝗅 *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.

ʃ

ʃφl *c1* sun

b

boφ *c2* food (0) eats (1)

ᴅ

ᴅ(l) *a* start (modifying) inceptive, inchoative

ᴅφȷs *c0* weather

ᴅ𝗅𝗅 *a* loan (1) is borrowing (0) [from (2)] (modifying) (0)'s (*) (borrowed)

ᴅՈȷ *c0* the listener or reader, you

ᴅΩl *c3* bread

ȷ

ȷn^{φ} *a* after

ȷφᴅ *a* time when (0), (1) (modifying) (*) happened at the time (0)

ȷΩᴅ *c1* walk

φ

φlh *c0* morning

φnʅſ *c2* fish

φΩh *a* absence it is not true that (0)

ʅ

ʅblΩ *c1* summer solstice

ʅlΩ *c0* the speaker or writer, I

n

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

nΩ *a* sight (0) sees (1)

ɗ

ɗſᴅ *c5* mountain

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)