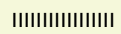




aaaaaaaaA, the language of *somewhere*

---

uruwi



aaaaaaaaaaaaaaaa

*A complete grammar*

15 November 2017

*Dedicated to Marek.*

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 . . . . .	3
1	Phonology and orthography . . . . .	5
1.1	Phoneme inventory . . . . .	5
1.2	Airflow . . . . .	5
1.3	Phonotactics . . . . .	6
1.4	Allophony . . . . .	6
1.5	The biting affix . . . . .	6
2	The statement space . . . . .	9
2.1	Conceptualisation . . . . .	9
2.2	Application . . . . .	10
2.3	Concrete inflections . . . . .	10
2.3.1	Season class . . . . .	10
2.3.2	Number mutability . . . . .	11
2.4	A simple example . . . . .	12
2.5	Modifiers . . . . .	13
2.5.1	Descriptors . . . . .	14
A	Dictionary . . . . .	17

## 0.1 | Introduction



# 1 | Phonology and orthography

## 1.1 | Phoneme inventory

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

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

Table 1.1: Phonemes of aaaaaaaaaA.  
Consonant

#	Hacm	(Egre)	(Ingr)	Vowel
0	ʃ	tʰ	!	eɪ
1	nʰ	ŋ	ɡ	ã
2	b	p	p	y
3	ɒ	m	ɓ	ũ
4	j	s	s	i
5	ɥ	j	j	i
6	ʟ	tʰ		ʌɪ
7	o	w	w	u
8	ɸ	k	k	o
9	l	l	l	u
10	ɭ	kʰ	‡	oɪ
11	n	n	n	eɪ
12	ɹ	r	ɹ	ɛ
13	ɹ	t	t	e
14	h	h	h	a
15	s	‡	‡	ʌ

When pronounced ingressively, the tones of vowels are inverted. That is, [ʌɪ↑] becomes [ʌɪ↓].

## 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) → /tʰ/ (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:

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

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

Thus, for instance, /hswlmŋ/ would be resolved to [hswpŋ], which could, for instance, be pronounced [asupā].

### 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( $\delta$ ) phonemes of  $\alpha$ , and xor them with  $\delta$  itself using the indices of the phonemes.
- In addition, switch the airflow at the start of the altered phonemes.
- Then append  $\omega$ .
- Resolve any phonotactic violations.

For instance, if we wanted to use  $\langle -\text{y:fn} \rangle$  on  $\langle \text{bɒl} \rangle$ , then we would:

- Take the last letter of  $\langle \text{bɒl} \rangle$ , namely  $\langle -l \rangle$  (9), and xor it with  $\langle -\text{y} \rangle$  (5).  $9 \vee 5 = 12$  so we now have  $\langle -r \rangle$ .
- Append  $\langle -fn \rangle$ . We now have  $\langle \text{bɒ} \rangle \text{rfn}$ .





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

$$X(A_1, \dots, A_n, C_1, \dots, C_n) \quad (2.1)$$

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

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

$$\text{Topic Operator Arguments} \dots \equiv \text{Operator}(\text{Topic}, \text{Arguments}) \quad (2.2)$$

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

$$\text{Operator Arguments} \dots \equiv \text{Operator}(\text{Arguments}) \quad (2.4)$$

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:

$$X \ulcorner \text{Arguments} \dots \lrcorner \equiv X(\text{Arguments}) \quad (2.5)$$

Inside the topic, the following is used instead:

$$\lrcorner \text{Arguments} \dots X \urcorner \equiv X(\text{Arguments}) \quad (2.6)$$

At the end of the sentence, any number of  $\lrcorner$ 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.

$\triangle$	js
$\ulcorner$	->q:fn
$\lrcorner$	b
$\lrcorner$	$\varphi$
$\urcorner$	->rl: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)			Name	Archetypes
#	Start	End		
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
$\Omega$	Multiversal time-constant	The quantity cannot change <i>under any circumstances</i> , or quantity is meaningless or irrelevant in this context.
$\Psi$	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.
$\Xi$	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).
$\Phi$	Continuously mutable	The quantity is continuously changing across time.
$\tilde{\Omega}$	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.
$\Sigma$	Situational	(in programming) The quantity might depend on the implementation.

Table 2.5: List of number mutability affixes.

S#	Affix 1	Sym	Affix 2
1	$\rightarrow n^{\Phi}$ :	$\Omega$	lbD
2	$\rightarrow j$ :	$\Psi$	lOn <sup>o</sup>
3	$\rightarrow l$ :	$X$	bOj
4	$\rightarrow b$ :	$Z$	lſy
5	$\rightarrow n$ :	$\Xi$	DyD
0	$\rightarrow h$ :	$\Phi$	nSl
		$\vec{\Omega}$	lbY
		$\vec{\Psi}$	lhd
		$\vec{X}$	bhl
		$\vec{\Xi}$	Dly
		$\vec{\Phi}$	nſd
		$\Sigma$	ln <sup>o</sup> 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:

- $\langle \text{bo}\phi \rangle$  c2 (0) eats (1)
- $\langle \text{fn}l \rangle$  c2 fish

- $\langle \Omega_{sn} \rangle$  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  $\Xi$  in this example. We use the same mutability for flowers.

Fronting the topic, we get:

ʎɤn/ɔɥɥ bɔɹɪɥɥ ɒs/ɔɥɥ  
 ɤnɪ-ɔɥɥ bɔɤ-ɔɥɥ ɒsn-ɔnʔ:ɔɥɥ  
 fish-2-ᐃ eat-2-ᐃ flower-1-ᐃ  
 Fish eat flowers.

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

\ֆո/յԺԿԸ boֆ ռՏՆ  
 ֆո/հ-յԺԿԸ boֆ ռՏՆ  
 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:

\p \p fo/lbaj d\p\j\nd\sbh \p \b\p\lbaj j\ndbh \p\j\ndh  
 \p \p \p\l->\nd\sbaj d\p\j\sb->\nd\sbaj \p \b\l->j\sbaj j\nd->\nd\sbaj \p\h->\nd\sbaj  
 \l \l sun-1-\tilde{X} weather-0-\Phi-\l \l summer solstice-2-\tilde{X} after-\l morning(\tilde{X})-\l

The rest of the sentence is thus:

ʃɔ ʁ/bɔɣʌlɪn ɪn ns/nʰn dʌʃɔɣ b ɔɪ  
 ʃɔ ʁɔ-ʔnʰ:ɔɣɔ-ʔɣɪn ɪn nɣ-ʔɣɪn dʌ-ʔn:ɔɣ b ɔɪ  
 time walk-1-ᑭ- speaker(ᑭ) destination- mountain-5-ᑭ 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 “ $\lceil \_ \rceil$ ”-overhead that usually applies. In this form, the direction of airflow is switched at the beginning of the descriptor:

$$\begin{aligned} \langle \phi | h \rangle \text{Morning} &\rightarrow \langle \rangle \phi | h \rangle D(\text{Morning}) \\ \langle j \phi \triangleright \rangle \text{Time} &\rightarrow \langle \rangle j \phi \triangleright \rangle D(\text{Time}) \end{aligned}$$

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

$\backslash \phi \triangleright \phi j / n^{\circ} \Omega s \rfloor j n^{\circ} \backslash \phi | \wedge b l \alpha \phi \backslash n^{\circ} \triangleright h$

ֆ ԾփյՏ→հ:ճՏՆ յո՞ >ֆ| >ՆԵՁ ֆԻհ→դ:Եհ  
 Լ weather-0-Փ after D-sun D-summer\_solstice morning-<sup>1</sup>

յֆԾ յՁ/ԵԾԿ\ԼԻ՛ն ոյ /ԻՁ \ԻԼԾ /ԵԼ՝  
 յֆԾ յՁԾ→ո՞:ԾԿԾ→Կ:Ին ոյ >ԻՁ >ԻԼԾ >ԵԼ՝  
 time walk-1-Է-<sup>1</sup> destination D-speaker D-mountain D-start





## A | Dictionary

---

| ʃ

ʃɸ| c1 sun

| b

boɸ c2 (0) eats (1)

| ɖ

ɖl| a start

ɖɸʃs c0 weather

ɖɸl c3 bread

| ʝ

ʝn<sup>ɸ</sup> a after

ʝɸɖ a time after (0), (1)

ʝɸɖ c1 walk

| ɸ

ɸlh c0 morning

ɸnɸ c2 fish

| ɿ

ɿblɿ c1 summer solstice

ɿlɿ c0 the speaker, I

| n

nɸ a destination to (0)

| ɳ

ɳfɖ c5 mountain

| ɳ

ɳsn c1 flower