
lel-ʼŋɪɸɔ-ɗɪ ɟɪ lel-ɗɪ ɟɪ jeʷle-ʼɸcʰɗɪɔ

aaaaaaaaA, the language of *Rymako*

uruwi

|||||

aaaaaaaaaaaaaa

A complete grammar

September 17, 2017

Dedicated to Isoraķatheð.

Branch: canon

Version: 0.1

Date: 2017-09-12

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

| Contents

0.1	Introduction	5
1	Phonology and orthography	7
1.1	Phoneme inventory	7
1.2	Hacmisation	8
1.3	Phonotactics	9
1.4	Stress	9
1.5	Vowel harmony	9
2	Syntax	11
2.1	Basic word order	11
2.2	Questions	11
2.3	Multiple clauses	11
3	Nouns	13
3.1	Number	13
3.2	Case	13
3.3	Noun classes	14
3.3.1	Countable	14
3.3.2	Measurable	14
3.3.3	Uncountable	14
3.4	Definiteness	14
3.5	Declension table	14
3.5.1	Countable classes	14
3.5.2	Measurable classes	15
3.5.3	Uncountable classes	15
3.6	Pronouns	16
3.6.1	Last-clause pronouns	17
3.7	Compounding	17
3.8	Possession	18

4	Verbs	19
4.1	Aspect	20
4.2	Obliques	22
4.3	Conjunctions	22
4.4	Subordinate clauses	23
4.5	Ditransitive-like constructions	23
4.6	Transitivisation	23
4.7	Clauses with nullary arguments	24
4.8	The copula	24
5	Descriptors	25
5.1	Conversion	25
6	Tree mode	27
6.1	Activation	27
6.2	Branch-switching	27
6.3	Anaphoric pronouns in joiner clauses	27
6.4	Errors	27
6.5	Example	28
7	Numerals	29
7.1	Single-digit numerals	29
7.2	Numerals up to $19 \cdot 17$	30
7.3	Numerals up to $13 \cdot 19 \cdot 17 = 4199$	30
7.4	Numerals up to and including $4199 \cdot (4199 + 1)/2 = 8817900$	31
7.5	Higher numerals	32
7.6	Cardinal and ordinal numerals	32
8	Names	33
8.1	Nominal names	33
8.2	Clausal names	34
9	Calendar	35
9.1	Tides	35
9.2	Months	37
9.3	Years	37
9.4	Eras	37
9.5	Subdivisions of the day	38
9.5.1	Traditional timekeeping	38
9.5.2	Modern timekeeping	38
	Listings of programs	41
9.6	workfiles/7/tides.sage	41
9.7	workfiles/7/bins.pl6	42
9.8	workfiles/7/conno.pl6	43
9.9	workfiles/7/count-days.pl6	44

0.1 | Introduction

1 | Phonology and orthography

1.1 | Phoneme inventory

Table 1.1: The consonants of aaaaaaaaaA.

	Bilabial	Alveolar	Palatal	Velar	Glottal
Nasal	m	n	ɲ	ŋ	
Plosive	p b	t d	c ɟ	k ɡ	ʔ
Fricative	f	s	ʃ	x	
(coarticulated)	θ x	ɸ x		ɸʃ	
Affricate		ts	tʃ		
Lateral fricative		ɬ			
Approximant		ɹ	j	w	
Lateral approximant		l			
Trill		r			

Table 1.2: The vowels of aaaaaaaaaA.

Spread	Half-rounded	Rounded
i	ɤ	y
u	ʊ	u
ɛ		œ
ʌ		ɔ
ä		

In addition to consonants and vowels, aaaaaaaaaA has rod signals, represented by numbers. Rod A is blue and held by one's dominant hand and B is red and held by one's non-dominant hand.

1. Rod A is raised to one's chest, while B is pointed down.
2. Rods A and B are crossed in the front.
3. Rod B is raised upwards in front of the nondominant arm, while rod A is lowered.

4. Rod A is pointed sideways near one's nondominant arm, while rod B is lowered.
5. Rods A and B are extended to the sides.
6. Rods A and B are extended, facing forward.
7. Rod A is raised forward, while B is pointed to the side.
8. Rod B is raised forward, while A is pointed to the side.

Lowering both rods is interpreted as an absence of a rod signal.

If the use of rods are unavailable, the numerals of the positions may be pronounced.

1.2 | Hacmisation

aaaaaaaaA uses the hacm script with superscript letters to indicate phonemes not found in Arka. The transcriptions can be found in Tables 1.3 and 1.4.

Table 1.3: The consonants of aaaaaaaaaA.

	Bilabial	Alveolar	Palatal	Velar	Glottal
Nasal	ɒ	n	n ^ɥ	n ^ɣ	
Plosive	d b	f ɳ	f ^ɥ ɳ ^ɥ	ɭ ɸ	.
Fricative	ɑ	j	l	h	
(coarticulated)	j ^h	ɑ ^h		ɑ ^l	
Affricate		fj	ɽ		
Lateral fricative		s			
Approximant		ɸ	ɥ	o	
Lateral approximant		l			
Trill		ɽ			

Table 1.4: The vowels of aaaaaaaaaA.

Spread	Half-rounded	Rounded
ɕ	ɕ ^ə	ɕ ^ɔ
ə ^ɕ	ə	ə ^ɔ
e		e ^ɔ
ɔ ^e		ɔ
ɪ		

Rod signs are represented by the hacm digits <1 2 3 4 5 6 7> attached to the end of the verbs they encompass. Proper words are preceded by a backslash <\>.

Vowels that are inferrable from context are sometimes omitted. For example, /æ-fan/ (to speak) is written <ɸeɑn>, but /æfin/ (to spread), which is less common, is written <ɸeɑcn>, with the second vowel. Most of this grammar will leave all vowels written.

1.3 | Phonotactics

An onset consists of one of the following:

- any single consonant other than /l/ (the exceptions are <le> [lek] and related words),
- any obstruent followed by an approximant other than /l/,
- or any plosive followed by /r/,
- or any nasal followed by /j/ or /w/.

A nucleus consists of one vowel.

A coda consists of one of the following:

- nothing,
- a nasal,
- a voiceless plosive (excluding /ʔ/),
- /ɹ/, /s/ or /l/

1.4 | Stress

Stress falls on the last syllable with a coda, or otherwise the second-to-last syllable.

See table 1.5 for examples.

Table 1.5: Examples of stress locations.

Orthography	Location of stress (# from last)
Dɪʃ	2
nɪ.cn	1
.əʔfələ	2
lɪjnedc ^a	3

1.5 | Vowel harmony

For the purposes of vowel harmony, vowels are divided into front and back vowels. /a/ is neutral. A root with neither front nor back vowels acts as if it has front vowels.

2 | Syntax

2.1 | Basic word order

The basic word order is VSO. Descriptors follow what they modify.

2.2 | Questions

Binary questions have the interrogative polarity marker and no change to syntax.

In wh-questions, the wh-word is pulled to the front (i. e. before the verb). This requires case marking for the wh-word:

flen peac^əjhi nc^ʔɕ
who-ACC speak-FAR.PAST-Q PR.FAR.SG
Whom did you speak to?

This applies only to questions, not interrogative-mood clauses that act as relative clauses:

peac^əjhi nc^ʔ flel, qif mu.
speak-FAR.PAST-Q PR.FAR.SG who, see-NEAR.PAST PR.ANAPH_OBJ.INT
I saw the person whom you talked to.

2.3 | Multiple clauses

A sentence might have multiple clauses. Each clause in a sentence follows the basic VSO order, and clauses are separated with commas.

3 | Nouns

Nouns are declined for number, case and definiteness.

3.1 | Number

aaaaaaaaA has many grammatical numbers:

Table 3.1: The discrete grammatical numbers of aaaaaaaaaA.

Number	Constraint on $x \in \mathbb{Z}$
Integral	none
Nullary	$x = 0$
Singular	$ x = 1$
Dual	$ x = 2$

Table 3.2: The continuous grammatical numbers of aaaaaaaaaA.

Number	Constraint on $x \in \mathbb{R}$
Nullary	$x = 0$
Subsingular	$ x < 1$
Supersingular	$1 \leq x < 2$
Plural	$ x \geq 2$ or x is unknown

3.2 | Case

In a clause with both the subject and object directly expressed in that order, both the subject and object are declined in the nominative case (and their roles are inferred through word order). In a clause where only one is present, or where both are expressed in the opposite order, the subject will receive the nominative case and the object will receive the accusative case.

3.3 | Noun classes

There are three overarching groups of noun classes.

3.3.1 | Countable

Nouns in these classes are declined for a discrete number.

1. Sentient – such as humans, AIs, deities.
2. Animate – nonsentient animals.
3. Inanimate – anything else.

3.3.2 | Measurable

Nouns in this class are declined for a continuous number.

4. Measure – all measurable nouns, especially units of measurement.

3.3.3 | Uncountable

Nouns in these classes are not declined for number, and require compounding with a countable or measurable noun in order to be quantified.

5. Fluid – liquids and gases.
6. Edible – edible (to humans) non-fluids.
7. Inedible – inedible (to humans) non-fluids.
8. Abstract – abstract ideas.

3.4 | Definiteness

The definite form of a noun is formed regularly by reduplicating the first syllable (without the coda): <𐤃𐤓𐤕> “a person” becomes <𐤃𐤓𐤃𐤓𐤕> “the person”.

3.5 | Declension table

3.5.1 | Countable classes

Note that noun declensions respect vowel harmony. For nouns with back vowels, replace the front vowels with the back vowels of the same height and rounding, and vice versa.

Table 3.3: Declensions for countable nouns.

	Integral	Nullary	Singular	Dual
--	----------	---------	----------	------

	Integral	Nullary	Singular	Dual
Sentient: <ɖɪɪ> “person”				
Nominative	ɖɪɪ	ɖɪɪle	ɖɪɪ	ɖɪɪl
Accusative	ɖɪɪn	ɖɪɪnʰle	ɖɪɪncɟ	ɖɪɪnɪl
Sentient: <ɟʰi.en> “magician”				
Nominative	ɟʰi.en	ɟʰi.eʃe	ɟʰi.eɟ	ɟʰi.el
Accusative	ɟʰi.eɲcn	ɟʰi.enʰle	ɟʰi.eɲcɟ	ɟʰi.eɲcl
(Note that the final consonant is preserved only in the integral nominative form.)				
Animate: <ɖənʰɔ> “rabbit”				
Nominative	ɖənʰɔ	ɖənʰɔɔʰe	ɖənʰɔɟ	ɖənʰɔ.əʰc
Accusative	ɖənʰɔn	ɖənʰɔnʰe	ɖənʰɔn	ɖənʰɔnəʰc
Animate: <ɟcʰɪ> “fox”				
Nominative	ɟcʰɪ	ɟcʰe	ɟcʰɟ	ɟcʰ.c
Accusative	ɟcʰɲcn	ɟcʰnʰle	ɟcʰɲcɟ	ɟcʰɲc
Inanimate: <ɰɪɾəʰ> “statue”				
Nominative	ɰɪɾəʰ	ɰɪɾəʰɔʰe	ɰɪɾəʰɪ	ɰɪɾəʰɟ
Accusative	ɰɪɾəʰɔ	ɰɪɾəʰɔɔʰe	ɰɪɾəʰɔʰe	ɰɪɾəʰəʰc
Inanimate: <.cɖen> “house”				
Nominative	.cɖen	.cɖeʃe	.cɖel	.cɖeɟ
Accusative	.cɖeɲcɔ	.cɖeɲcɔʰe	.cɖeɲcɔʰe	.cɖeɲcac

3.5.2 | Measurable classes

Table 3.4: Declensions for measurable nouns.

	Plural	Nullary	Subingular	Supersingular
Measure: <ɲəɖəʰ> “day (continuous)”				
Nominative	ɲəɖəʰ	ɲəɖəʰɲəʰ	ɲəɖəʰɪ	ɲəɖəʰn
Accusative	ɲəɖəʰn	ɲəɖəʰɲəʰn	ɲəɖəʰnʰe	ɲəɖəʰnəʰn
Measure: <ɖel> “volume” (in expressions such as <ɖel-ɲəɟəʰ> “cupful”)				
Nominative	ɖel	ɖeɲcʰ	ɖeɲcɪ	ɖeɲcn
Accusative	ɖeɲcn	ɖeɲcʰn	ɖeɲcnʰe	ɖeɲcnɲcn
(Note that the final consonant is preserved only in the plural nominative form.)				

3.5.3 | Uncountable classes

Notably, uncountable-class noun declensions do not respect vowel harmony.

Table 3.5: Declensions for measurable nouns.

	Mass
Fluid: <ɑʰəɲəʰ> “water”	
Nominative	ɑʰəɲəʰ
Accusative	ɑʰəɲəʰn
Fluid: <neled> “nitrogen”	

ment becomes the first.

If a clause has no explicit arguments, the first argument is understood to be the subject.

Table 3.7: Personal pronouns. <-n>, <-en> or <-ɔ̃n> is suffixed for the accusative case.

(continuous) (discrete)	Pl. / Sub. / Sup. Integral	Nullary Nullary	Singular	Dual
Near	fi	lefi	de	acɥc
Far	ɔ̃	ɔ̃de	nə̃	bɥi
Other	nc	lenc	sc̃	ɔ̃ɥc
Anaph. Sub.	ɥi	leɥi	.cɔ̃	ñɥcɥc
Anaph. Obj.	ɥɔ̃	ɔ̃ɥɔ̃	.ə̃c̃	ñɥə̃c̃ɥɔ̃
Generic	.ə̃			

(For the observant readers: notice the similarity to Kavinan’s system.)

3.6.1 | Last-clause pronouns

The anaphoric pronoun <bej> (accusative: <bejen>) is grammatically an other pronoun, and it refers to the previous clause said. Likewise, <bedcj> (accusative: <bedcn>) refers to the clause before the previous one.

3.7 | Compounding

Nouns can be compounded together in a head-initial manner. When that happens, only the leftmost noun is the one to be declined.

ɔ̃el-ɥəɔ̃-ə̃l̃ə̃ɥə̃-ɥcɥ
 volume-cup-water-five
 five cupfuls of water

Note that integral pronouns can modify other nouns, in which personal possession is indicated:

ɔ̃el-ɥəɔ̃-ə̃l̃ə̃ɥə̃-ɥcɥ-fi
 volume-cup-water-five-PR.NEAR.INTEGRAL
 (arg1)’s five cupfuls of water

Descriptors can also compound on nouns. This compounding is productive in aaaaaaaaA.

ɔ̃ɥi-l̃ə̃fi
 person-old
 old people
 (Compare to ɔ̃ɥi l̃ə̃fi “person old-SENTIENT”.)

3.8 | Possession

“X’s Y” is translated as $\langle Y=D_l \ jcl \ X \rangle$. The possessive construction is also used to create appositives.

Observe that possession marks the head, and $\langle -D_l \rangle$ is a clitic, not an affix, as in the following example:

$dəbən^wəj - a'əpə - D_l \ jcl \ j^h i.ej$
 DEF~rabbit-SING-water=GEN POS magician-SING
 the magician's water rabbit

In more casual speech, $\langle jcl \rangle$ may be dropped.

4 | Verbs

Verbs are conjugated for person of the subject, tense, polarity and tellicity, in two paradigms. Conjugation respects vowel harmony.

Table 4.1: Person-tense conjugations for verbs, using <ḍilɪn> “(S) eats (O)”.

	Nonpast	Past
Near	ḍilɪn	ḍilɪf
Far	ḍilɪn	ḍilc ^a j
Other	ḍilɪ	ḍilc ^a
Anaph. Sub.	ḍile	ḍilel
Anaph. Obj.	ḍilc.e	ḍilc.el
Generic	ḍilc ^ɔ	ḍilc ^ɔ

Table 4.2: Person-tense conjugations for verbs, using <peacn> “(S) spreads (O)”.

	Nonpast	Past
Near	peacn	peacf
Far	peaɪn	peac ^a j
Other	peaɪ	peac ^a
Anaph. Sub.	peae	peael
Anaph. Obj.	peac.e	peac.el
Generic	peac ^a	peac ^a

to which a suffix is added:

Table 4.3: Polarity-tellicity suffixes for verbs. The interrogative affix can also follow a negative affix.

	Positive	Negative	Interrogative
Telic	–	–le / –lɔ ^e	–hɪ
Atelic	–ɔc / –ɔc ^c	–ɪ	–lc ^a / –lə

Notes:

- “Negative atelic” means something akin to “unsuccessfully tried to avoid doing X”.
- The interrogative polarity, in addition to marking questions, is used to mark clauses that may or may not be true but are referred to later in the sentence.

Some examples:

ḏḏḏḏ ḏḏḏḏ ḏḏḏḏ.
eat-NEAR.NONPAST fish flower
Fish eat flowers.

ḏḏḏḏ ḏḏḏḏ ḏḏḏḏ, ḏḏḏḏ ḏḏḏḏ ḏḏḏḏ.
eat-NEAR.NONPAST fish flower, eat-NEAR.NONPAST cat PR.ANAPH_SUB
Fish eat flowers, and cats eat fish.

ḏḏḏḏ ḏḏḏḏ ḏḏḏḏ, ḏḏḏḏ ḏḏḏḏḏḏ.
eat-NEAR.NONPAST fish flower, eat-ANAPH_SUB.NONPAST grass-ACC
Fish eat flowers, and they eat grass.
(Grass is inedible to humans, but edible to fish.)

ḏḏḏḏḏḏ ḏḏḏḏ ḏḏḏḏ.
eat-NEAR.NONPAST-NEG flower fish
Flowers don’t eat fish.

ḏḏḏḏ ḏḏḏḏ ḏḏḏḏḏḏḏ, ḏḏḏḏ ḏḏḏḏ.
carry-OTHER.NONPAST PR.OTHER.SG DEF~book-SG, worry-NEAR.NONPAST PR.NEAR.INT
PR.LAST_CLAUSE
He has the book; that worries me.
or: That he has the book worries me.

ḏḏḏḏḏ ḏḏḏḏ ḏḏḏḏḏḏḏ, ḏḏḏḏ ḏḏḏḏ.
carry-OTHER.NONPAST-INTERROGATIVE PR.OTHER.SG DEF~book-SG, worry-NEAR.NONPAST
PR.NEAR.INT PR.LAST_CLAUSE
He might have the book; that worries me.
or: That he might have the book worries me.

4.1 | Aspect

Verbs can also be marked for aspect, either using a rod sign directly on the verb, or a particle with a rod sign, placed anywhere between the verb it modifies and the next verb.

Table 4.4: Aspect markers. Those with hyphens are attached to verb. Those without hyphens are placed as separate particles anywhere after the verb.

Aspect name	Marking	Meaning
Imperfect	–ɿ	An action that is currently going on. Also used to distinguish static actions as opposed to dynamic (e. g. <i>wear</i> as opposed to <i>put on</i>).
Interrupted	ɿcɿ	An action that was interrupted.
Perfect	–ɿ	An action that has already finished. Changes present tense to immediate past. Also used to distinguish dynamic actions as opposed to static (e. g. <i>put on</i> as opposed to <i>wear</i>).
Gnomic	–ʔ	A general truth or aphorism, or an action done habitually.
Gnomic dubitative	ɿcɿʔ	A general truth or aphorism that the speaker considers to be false.
Deontic necessity	–ɹ	An action that the speaker insists on happening.
Epistemic necessity	ɿəᵈɹ	An action that the speaker infers that is happening.
Deontic potential	–ʔ	An action that the speaker permits to occur.
Epistemic potential	ɿəᵈʔ	An action that the speaker infers that might happen.
Unexpected	–ɿ	An action that is unexpected (akin to using “but”).
Comparative	deɿ	Indicates an action of greater intensity than what was described in the previous clause.
Nonexclusive subject	ɿcɿ	Indicates that the subject comprises not only of what is explicitly mentioned, but also other things.
Nonexclusive object	ɿcʔ	Indicates that the object comprises not only of what is explicitly mentioned, but also other things.
Nonexclusive argument	ɿcɹ	Combination of both nonexclusive subject and nonexclusive object.

An example:

ɿɿɿɿɿcɿ de nc, lcnc.ɛɿ dənʔɿdɔᵈ–ɿ.
 fight-NEAR.PAST-ATELIC-IMPERFECT PR.NEAR.SG PR.OTHER.INT, shoot-ANAPH_OBJ.PAST-
 UNEXPECTED knee-SG.ACC-PR.ANAPH_SUB.INT
 I tried to fight them, but they shot my knee.

4.2 | Obliques

aaaaaaaaA lacks oblique arguments. Instead, equivalent expressions employ serial verb constructions. For instance, “he ate soup with a spoon” would be reduced to “he held a spoon and ate soup”:

dɿc^a sc^o ɔ^olɔ^ol, ɒilɛl sɪdʒɒn.
 INST-OTHER.PAST PR.OTHER.SG spoon-SG, eat-ANAPH_SUB.PAST soup-ACC
 He held a spoon and ate soup.
 or: He ate soup with a spoon.

Likewise:

nɪ.ɪ nc jəjəl-hɪ.ɪp, ncɒɛ hɪhɪɾə^odɔ^o.
 TEMPORAL-OTHER PR.OTHER.INT DEF~day-SG-spring, dance-ANAPH_SUB DEF~statue-SG-ACC
 They will wait until the spring equinox and dance around the statue.
 or: They will dance around the statue on the spring equinox.

A similar construction can be used for the negation of obliques:

dɿc^ale sc^o ɔ^olɔ^ol, ɒilɛl sɪdʒɒn.
 INST-OTHER.PAST-NEG PR.OTHER.SG spoon-SG, eat-ANAPH_SUB.PAST-UNEXPECTED soup-ACC
 He did not hold a spoon, but ate soup.
 or: He ate soup without a spoon.

4.3 | Conjunctions

Conjunctions such as “and” are treated like obliques. For instance, “and” is represented by the verb <acn>, and precedes the clause in which the two are used:

ac^a \ɿc^aje \ɪpəl, ɒilɛl fɛp.c^on.
 and-OTHER.PAST Ryse Tarul, eat-ANAPH_SUB.PAST beef-ACC
 Ryse and Tarul ate beef.

Sufficiently complex nesting may be unrepresentable using only anaphoric referents. The easiest way to resolve this issue is to use definite nouns in place of anaphoric referents.

ac^a \ɿc^aje \ɪpəl, ac^a fɛp.c^o sɪdʒɒ, ɒɪlc^a ɒɪɒɪɪl ɱɪ.
 and-OTHER.PAST Ryuse Tarul, and-OTHER.PAST beef soup, eat-OTHER.PAST DEF~person-DU PR.ANAPH_SUB.CONT
 [They,] Ryse and Tarul ate beef and soup.

4.4 | Subordinate clauses

Ideas such as “if” or “because” are also expressed with verbs. For example, <ni.cn> “wait, when” is also used for “if”:

feɸilehi, ni.cn fi bej, aehcn.
 rain-OTHER-NEG-Q, wait-NEAR PR.NEAR.INT ANAPH_CLAUSE, play-NEAR
 If it doesn't rain, we will play.

Note the clausal argument to <ni.cn>, since our condition is an entire clause instead of a noun.

4.5 | Ditransitive-like constructions

In English, some verbs such as *give* take two objects: the item being given and the recipient of the item. Since clauses in aaaaaaaaA can take only one object, translating such verbs requires multiple clauses:

fiɸif de hɸchɸcæn, nebel ɸc^əjen.
 lose-NEAR.PAST PR.NEAR.SG DEF~book, give_to-ANAPH_SUB.PAST Ri^use-ACC
 I gave the book to Ryse.

4.6 | Transitivity

Verbs that are intransitively (i. e. have no object passed at this time) can be turned into a causative form with the prefix <ɸc->:

ɸcɸɸɸɸ aɸeapen^ue.
 fall-NEAR.PAST DEF~coin
 The coins fell.

de ɸcɸɸɸɸ^ə aɸeapen^ue
 PR.NEAR.SG TRANS-fall-OTHER.PAST DEF~coin
 I dropped the coins.

Note that the word order changes to SVO. In addition, the verb is conjugated for its object, rather than the subject as expected. If the following clause uses an anaphoric subject, it refers to the object of the current clause.

Moreover, the verb does not need to be one that can never take an object. In the above example, <ɸcɸɸɸɸ> means “(S) falls on (O)”. However, if the verb in question is taking an object, it cannot be transitivised directly and a more roundabout way is required:

ɸcɸɸɸɸ aɸeapen^ue ɸifi.
 fall-NEAR.PAST DEF~coin grass

The coins fell on the grass.

De ꝥcꝥcꝥc^a aꝥeꝥeꝥe^e, ꝥcꝥeꝥ ꝥꝥꝥe.

PR.NEAR.SG TRANS-fall-OTHER.PAST DEF~coin, fall-ANAPH_SUB.PAST grass-ACC

I dropped the coins; they fell on grass.

or: I dropped the coins on grass.

4.7 | Clauses with nullary arguments

A clause with one or more arguments that are nullary or modified by nullary-number nouns (either through compounding or possession) will have a negative verb as well:

Dæꝥꝥe ꝥꝥꝥe.

recall-NEAR-NEG person-NULL

No one knows.

ꝥꝥꝥꝥe De ꝥꝥꝥꝥ ꝥꝥ ꝥꝥe.

want-NEAR-NEG PR.NEAR.SG ring=GEN POS magician-NULL

I don't want the rings of any magician.

4.8 | The copula

The copula <jcn> can take a noun as an object, in which case it can mean identity or membership. (Location is expressed with <ꝥꝥ> “be at”.) With no object at all, it is used to denote existence.

It can also accept a descriptor, in which case the descriptor is attached before <jcn> in the dictionary form.

5 | Descriptors

Descriptors act as adjectives or adverbs. They follow what they modify, and are inflected for the noun class or verbal person of their antecedents.

Table 5.1: Descriptor declensions, using the descriptors <hedfi> “large” and <leʔfi> “old”.

Class or person	Declined form	
Sentient	hedfi	leʔfi
Animate	hedfi	leʔfi
Inanimate	hedfe	leʔfə ^e
Measure	hedfiy	leʔfiy
Fluid	hedfej	leʔfə ^e y
Edible	hedfc	leʔfə ^c
Inedible	hedfeʔ	leʔfə
Abstract	hedfc ^a	leʔfə
Near	hedfiy	leʔfiy
Far	hedfiy	leʔfiy
Other	hedfey	leʔfey
Anaph. Sub.	hedfiy	leʔfiy
Anaph. Obj.	hedfey	leʔfey
Generic	hedfc ^y	leʔfə ^y

5.1 | Conversion

A noun can be converted to a descriptor by appending <-i>.

A descriptor can be converted to an abstract noun meaning “the nature of being ~” by replacing the final <-i> with <-cnel>.

6 | Tree mode

As mentioned in section 4.3, anaphoric referents in a linked-list sentence are sometimes insufficient for expressing even simple sentence structures. While the easiest method of resolving this issue is using definite nouns, aaaaaaaaaA also provides a mode where sentences are not linked lists of clauses, but rather (binary) trees.

6.1 | Activation

Tree mode is enabled automatically when the treeing particle <ᵐᵃᵐ> is used, and disabled at the end of a sentence.

6.2 | Branch-switching

The aforementioned particle <ᵐᵃᵐ> marks the beginning of the right branch of the tree. The right branch is ended by the particle <ᵐᵃᵐᵃ>, which causes the next clause to join the left and right branches.

(N. B. <ᵐᵃᵐ> and <ᵐᵃᵐᵃ> can occur only between clauses. If the particles are represented by left and right brackets, respectively, then the brackets should match.)

6.3 | Anaphoric pronouns in joiner clauses

In clauses that join two branches, anaphoric pronouns require marking whether the antecedent occurs in the left predecessor <ᵐᵃᵐ> or the right predecessor <ᵐᵃᵐᵃ>. This is done by marking the pronoun with <-ᵐ> or <-ᵐᵃ>.

Likewise, verbs can be modified with <-ᵐ> or <-ᵐᵃ> to indicate which branch the subject came from.

6.4 | Errors

The following are ungrammatical:

- Using the particle <ᵐᵃᵐᵃ> or the branched anaphoric pronouns when tree mode is disabled

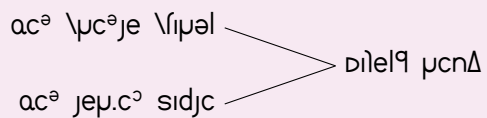
- Using the particle $\langle n^4\Delta \rangle$ other than to close a corresponding $\langle n^4\eta \rangle$
- Using the unbranched anaphoric pronouns in clauses with two predecessors
- Using the branched anaphoric pronouns in clauses with one predecessor
- Starting a new branch with $\langle n^4\eta \rangle$ when the current branch is empty

6.5 | Example

The second example in section 4.3 can be expressed as follows:

$ac^a \setminus pc^a\eta \setminus \eta\mu\theta, n^4\eta ac^a \eta\mu.c^\circ \text{ sidjc}, n^4\Delta \text{ d}\eta\eta\eta\eta \mu\text{cn}\Delta.$

The resulting tree is shown below:



7 | Numerals

aaaaaaaaA uses a mixed-base system for its numerals. Numerals are abstract nouns. A Perl 6 program to convert numerals can be found in Section 9.8.

7.1 | Single-digit numerals

Here are the numerals for $n < 17$:

Table 7.1: The cardinal numbers from 0 – 16.

base 10	base v	word
0	0	μəɒ
1	1	ac ^ə l
2	2	ʃɪj
3	ʔ	ɔ ^h e ^ə n
4	ŋ	ɑ ^h ɪμ
5	ʔ	ʔcɪ
6	ɿ	ɒɪe
7	ɹ	ʃɪcɪ
8	Δ	də ^ə n
9	L	hed
10	F	bən ^ə
11	7	nə ^c
12	£	le ^ə
13	‡	j ^h cd
14	A	ɪɪn
15	V	lel
16	ψ	.ɪμ

Note that digits above 9 use capital hacm letters.

7.2 | Numerals up to $19 \cdot 17$

These are represented by two digits. The multiples of 17 are shown below:

Table 7.2: Multiples of 17, up to $18 \cdot 17$.

base 10	base v	word
17	10	selc ^a
34	J0	ɔilc ^a
51	ʔ0	ɔ ⁴ e ³ lc ^a
68	00	ʃɔlc ^a
85	ʔ0	ʔcʃlc ^a
102	ʃ0	ɔ ⁴ elc ^a
119	90	ʃcʃlc ^a
136	Δ0	də ³ nə
153	L0	helc ^a
170	F0	bən ⁴ ə
187	70	nə ^c lə
204	£0	le ³ lc ^a
221	‡0	j ^h cdlc ^a
238	A0	yinc ^a
255	V0	lelyc ^a
272	ʔ0	.ilyc ^a
289	Ð0	jɪlsc ^a
306	‡0	heɔdc ^a

Thus $y \cdot 17 + x$ is written $\langle x-y \rangle$.

7.3 | Numerals up to $13 \cdot 19 \cdot 17 = 4199$

These are represented by three digits. The multiples of $19 \cdot 17$ are listed below:

Table 7.3: Multiples of $19 \cdot 17$, up to $12 \cdot 19 \cdot 17$.

base 10	base v	word
323	100	hɪjɪn
646	J00	ʃɪjɪn
969	ʔ00	ɔ ⁴ e ³ jɪn
1292	000	ɔ ⁴ ɪjɪn
1615	ʔ00	ʔcɪjɪn
1938	ʃ00	ɔ ⁴ eɪjɪn
2261	900	ʃcɪjɪn
2584	Δ00	də ³ nɪn
2907	L00	heɪjɪn
3230	F00	bən ⁴ ɪn

base 10	base v	word
3553	700	nə ^c jɪɪn
3876	£00	le ³ jɪɪn

Thus $(z \cdot 19 \cdot 17) + (y \cdot 17) + x$ is written $\langle z-x-y \rangle$.

7.4 | Numerals up to and including $4199 \cdot (4199 + 1)/2 = 8817900$

The numeral for 4199 is $\langle \text{ə.ɔlə}^c \rangle$, written as $\langle 1:000 \rangle$.

Likewise, two $\langle \text{ə.ɔlə}^c \rangle$ is written as $\langle \text{J}:000 \rangle$ and pronounced $\langle \text{ə.ɔlə}^c\text{-fɪj} \rangle$, but the second $\langle \text{ə.ɔlə}^c \rangle$ is one smaller than the first. In other words, $\langle \text{J}:000 \rangle = 4199 + (4199 - 1) = 8397$.

Table 7.4: “Multiples” of $\langle \text{ə.ɔlə}^c \rangle$.

“Multiple”	Difference from last	Total
(0)		0
1:000	4199	4199
J:000	4198	8397
ʔ:000	4197	12594
0:000	4196	16790
ʔ:000	4195	20985
...		
£.ʔV:000	3	8817897
£.ʔW:000	2	8817899
1::000:000	1	8817900

Thus the n th “multiple” differs from the $(n-1)$ th multiple by $(4199 + 1 - n)$ (given $1 \leq n \leq 4199$), and the sum of the first n “multiples” is

$$\begin{aligned}
 y(n) &= \sum_{i=1}^n (4200 - i) \\
 &= \frac{1}{2} \cdot (8399 \cdot n - n^2)
 \end{aligned} \tag{7.1}$$

And likewise, for some given y , the largest “multiple” of $\langle \text{ə.ɔlə}^c \rangle$ not smaller than y has the index

$$N(y) = \left\lceil \frac{1}{2} \cdot (8399 - \sqrt{70543201 - 8 \cdot y}) \right\rceil \tag{7.2}$$

In other words, for any numeral $\langle n_1:n_2 \rangle$, $n_1 + n_2$ must be less than 4199.

7.5 | Higher numerals

The bases of higher numerals b_i can be derived from the recurrence relation

$$b_i = \begin{cases} 4199 & \text{if } i = 1 \\ \frac{b_{i-1} \cdot (b_{i-1} + 1)}{2} & \text{otherwise} \end{cases} \quad (7.3)$$

Then b_i acts as a new triangular base. Equations 7.1 and 7.2 can be generalised to the following:

$$y(n, b_i) = \sum_{i=1}^n (b_i + 1 - n) \quad (7.4)$$

$$= \frac{1}{2} \cdot (n \cdot (2 \cdot b_i + 1 - n)) \quad (7.5)$$

$$N(y, b_i) = \left\lfloor \frac{1}{2} \cdot \left(2 \cdot b_i - \sqrt{4 \cdot b_i^2 + 4 \cdot b_i + 1 - 8 \cdot y} \right) \right\rfloor \quad (7.6)$$

It follows that $y(n_1, b_i) + n_2$ is represented as $\langle n_1 :^i n_2 \rangle$ (i colons), and such a numeral must satisfy $n_1 + n_2 < b_i$.

Here are the names of the bases themselves:

Table 7.5: Names of higher bases.

Base	base v	word
b_1	1:000	ṽə.ʒlə ^c
b_2	1::000:000	ɒɐɹc ^ʒ ṽ
b_3	1:::000:000::000:000	ṽṽṽɹə ^ʒ
b_4		ṽṽṽṽcɔṽ

7.6 | Cardinal and ordinal numerals

Cardinal numerals compound to their antecedents; ordinal numerals use the possessive $\langle -\text{Dṽ } \text{jcl} \rangle$ construction:

ḌCQI–ṽ^ʒṽn
 child-three
 three children

ḌCQIḌṽ ṽcl ṽ^ʒṽn
 child-SG=GEN POS three
 the third child

8 | Names

Names fall into two grammatical categories:

- *Nominal names* act as nouns. They are usually single words.
- *Clausal names* are entire clauses. These names usually refer to places, although a few people have clausal names. In extreme cases, such a name can span multiple clauses.

8.1 | Nominal names

These names act as nouns, and they are preceded by a backslash <\>. If the name spans multiple words (as common in foreign names), spaces are escaped by backslashes. No distinction is made between native and foreign names.

Only personal names can stand on their own, and even then, only given or full names. Other names must modify a common noun describing the nature of what is named, in the integral number without definiteness.

Table 8.1: Some examples of nominal names.

Name	Type
\pɕ ^ə je	Personal (native)
\ɸɪpəl	Personal (native)
\pɛɓcn	Personal (foreign)
dpe ^ɔ ɓe-\oɪɪ.c	Place (foreign)

Native names will usually respect vowel harmony. Children of parents who work in professions demanding physical labour (e. g. bricklaying) will usually have names with back vowels. In contrast, those born to parents of professions that do not demand physical strength (e. g. computer programming) will usually bear names with front vowels.

These names comprise of one or more clauses. Due to the nature of clausal names, they are all considered native. Most of these names refer to places; personal clausal names are almost always nicknames or such. Orthographically, they are put into square brackets <[]>.

We call the *referent* the subject, the object or the verb of the last clause, respectively depending on the type of anaphoric pronoun used to refer to the name. If the referent is a noun, it must be declined in the integral number without definiteness.

Table 8.2: Some examples of clausal names.

Name	Type	Literal meaning
[AEXʰEʰ dɾɪlɪjɪ ɾɔɔɔ]	Place	The trees covered the ground
[Dəɪ ʰXEʰDE ɲeʰ]–[ɫenɲeʰ]	Place	The city remembers the Šedry star
[aʰ ɪɲɪʰə–Dɪ jɪ jəɲ ɟʰɪ.en–Dɪ jɪ ɪɲeʰ, ʰoeɟe.el ʰXEʰDEʰ]	Place	The city was founded by the warrior of the sun and the wizard of the moon
[Deɪ ɪʰETʰ ɟeʰ]–[seɪʰ]	Personal	Gulto takes care of 17 foxes

ni.ɪ biːnən-bəj, [bəʃi dʲeːˈbe ɯːəʃ]-\lənɯːʃ, ʃʰi diːdiɭij .CD.
wait-OTHER year-future, (name), go-OTHER DEF~person-SG PR.ANAPH_SUB.SG
He will go to Muta Pröme Ryk-Šedrî next year.

9 | Calendar

Domain II, which contains *Rymako* has a day that is 26.99410 hours long. Other figures are given in terms of local days:

Table 9.1: Astronomical measures for Domain II.

Period	Length in local days
Local (synodic) day	1.00000
Sidereal day	0.99699
Tropical year (l_y)	301.94714
Sidereal year	302.03719
Synodic month (l_m)	30.80152
Sidereal month	27.95032

9.1 | Tides

In Domain II, the offset of the sea level due to the tide can be modeled by the following equations:

$$y = y_s + y_m \quad (9.1)$$

$$y_s = A_s \cdot (1 + A_{sa} \cdot \cos(\tau \cdot t)) \cdot \cos(2 \cdot \tau \cdot t) \quad (9.2)$$

$$y_m = A_m \cdot \left(1 + A_{ma} \cdot \cos\left(\frac{\tau \cdot t}{l_m}\right)\right) \cdot \cos\left(\frac{2 \cdot \tau \cdot (1 - l_m) \cdot t}{l_m}\right) \quad (9.3)$$

where:

$$\begin{aligned}
\tau &= 2 \cdot \pi \\
A_s &\approx 0.675 \\
A_{sa} &\approx 0.0532 \\
A_m &\approx 1.267 \\
A_{ma} &\approx 0.176 \\
y &= \text{offset of sea level in metres} \\
t &= \text{time since HAT in local synodic days}
\end{aligned}$$

An exact solution to $dy/dt = 0$ is not known to exist. However, the solutions to this equation can be found numerically. Consult Section 9.6 for a Sage program to do so.

As the calendar used by aaaaaaaaaA uses the high and low tides to count time, it is not synchronised even with days. The basic unit of time in the calendar is the *tidal day* $\langle \mu\theta\delta\theta^c \rangle$ (l_t) – the amount of time between a high tide and the second high tide thereafter, which is, on average, 1.03356 local synodic days, but can vary considerably. Thus:

$$l_m/l_t \approx 29.80148 \tag{9.4}$$

$$\approx 4053/136 \tag{9.5}$$

$$l_y/l_m \approx 9.80299 \tag{9.6}$$

$$\approx 7215/736 \tag{9.7}$$

This suggests that:

1. most months will have 30 days, but every 136 months, 27 months will have only 29.
2. most years will have 10 months, but every 736 years, 145 years will have only 9.

Figure 9.1: Table of year lengths in a cycle.

0123456789		0123456789		0123456789		0123456789
0 XXXXX9XXXX		19 XX9XXXX9XX		38 9XXXX9XXXX		57 XXX9XXXX9X
1 9XXXX9XXXX		20 XXX9XXXX9X		39 9XXXX9XXXX		58 XXX9XXXX9X
2 9XXXX9XXXX		21 XXX9XXXX9X		40 9XXXX9XXXX		59 XXX9XXXX9X
3 9XXXX9XXXX		22 XXX9XXXX9X		41 X9XXXX9XXX		60 XXX9XXXX9
4 9XXXX9XXXX		23 XXX9XXXX9X		42 X9XXXX9XXX		61 XXX9XXXX9
5 9XXXX9XXXX		24 XXX9XXXX9X		43 X9XXXX9XXX		62 XXX9XXXX9
6 9XXXX9XXXX		25 XXX9XXXX9X		44 X9XXXX9XXX		63 XXX9XXXX9
7 X9XXXX9XXX		26 XXX9XXXX9		45 X9XXXX9XXX		64 XXX9XXXX9
8 X9XXXX9XXX		27 XXX9XXXX9		46 X9XXXX9XXX		65 XXX9XXXX9
9 X9XXXX9XXX		28 XXX9XXXX9		47 XX9XXXX9XX		66 XXX9XXXXX
10 X9XXXX9XXX		29 XXX9XXXX9		48 XX9XXXX9XX		67 9XXXX9XXXX
11 X9XXXX9XXX		30 XXX9XXXX9		49 XX9XXXX9XX		68 9XXXX9XXXX
12 X9XXXX9XXX		31 XXX9XXXX9		50 XX9XXXX9XX		69 9XXXX9XXXX
13 X9XXXX9XX		32 XXX9XXXX9		51 XX9XXXX9XX		70 9XXXX9XXXX
14 XX9XXXX9XX		33 XXX9XXXXX		52 XX9XXXX9XX		71 9XXXX9XXXX
15 XX9XXXX9XX		34 9XXXX9XXXX		53 XX9XXXX9X		72 9XXXX9XXXX
16 XX9XXXX9XX		35 9XXXX9XXXX		54 XXX9XXXX9X		73 9XXXX9
17 XX9XXXX9XX		36 9XXXX9XXXX		55 XXX9XXXX9X		
18 XX9XXXX9XX		37 9XXXX9XXXX		56 XXX9XXXX9X		

9: 9 months

X: 10 months

9.5 | Subdivisions of the day

Lek-Tsaro has two systems for subdividing the day.

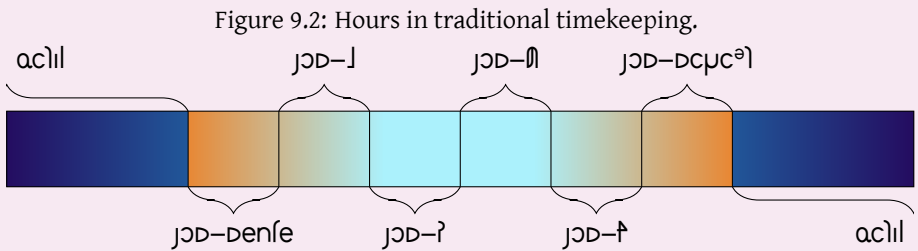
9.5.1 | Traditional timekeeping

The traditional system of timekeeping relies on subdivisions of the solar day. As shown in Figure 9.2, the period between sunrise and sunset are divided into six equally-sized parts <ᵛᵛᵛ>, and the night is considered a separate category. This implies that the length of the “hours” depends on the seasons.

The traditional timekeeping system is vague – typically, the most precise interval used is a quarter or eighth of an “hour”.

9.5.2 | Modern timekeeping

The need for precise schedules necessitated another standard for subdividing the day. The modern system is based on the tidal day, rather than the solar day. In theory, each



tidal day is divided into 23 equal parts $\langle \text{לח} \rangle$, each of which is divided into 80 equal parts $\langle \text{כנע} \rangle$, which are each divided into 40 equal parts $\langle \text{בדח} \rangle$.

Of course, having 23 $\langle \text{לח} \rangle$ per tidal day requires predicting the next two high tides. For that reason, each day's $\langle \text{לח} \rangle$ are based on the length of the *previous* tidal day, such that each day might have more or less than 23 $\langle \text{לח} \rangle$.

| Listings of programs

9.6 | workfiles/7/tides.sage

```
1 # How many values to output
2
3 limit = int(sys.argv[1]) if len(sys.argv) > 1 else 1000
4
5 # :P
6
7 tau = 2 * pi
8
9 t = var("t")
10
11 # Constants
12
13 A_s = 0.675; A_sa = 0.0532; A_m = 1.267; A_ma = 0.176; l_m = 30.80152
14
15 # Solar component
16 y_s2 = A_s * (1 + A_sa * cos(tau * t)) * cos(2 * tau * t)
17 # Lunar component
18 y_m2 = A_m * (1 + A_ma * cos(tau * t / l_m)) * cos(2 * tau * t / l_m -
19             2 * tau * t)
20 y = y_s2 + y_m2
21 yp = diff(y, t)
22
23 # High and low tides occur at values of t where dy/dt = 0.
24
25 i = 0
26 time = 0
27 print(0)
28 while i < limit:
29     try:
30         time2 = find_root(yp == 0, time + 0.000000001, time + 0.35)
31         print(time2)
32         time = time2
33         i += 1
34     except:
35         time += 0.01
```

workfiles/7/tides.sage

9.7 | workfiles/7/bins.pl6

```

1  # CONSTANTS
2
3  constant \MONTHS_PER_YEAR_CYCLE = 7215;
4  constant \YEARS_PER_YEAR_CYCLE = 736;
5  constant \AVG_MONTHS_PER_YEAR = MONTHS_PER_YEAR_CYCLE /
    YEARS_PER_YEAR_CYCLE;
6
7  # COMPUTATION
8  # For each year, take as many months as are needed
9  # in order to cycle to the next.
10
11 my $c = 0;
12 my @k;
13
14 for 0 ..^ YEARS_PER_YEAR_CYCLE -> $i {
15     my $need = 1 - ($c - floor($c));
16     my $objs = ceiling($need * AVG_MONTHS_PER_YEAR);
17     @k[$i] = $objs;
18     $c += $objs / AVG_MONTHS_PER_YEAR;
19 }
20
21 # DISPLAY
22
23 my \cols = 4;
24 my $len = @k.elems;
25
26 say ("    0123456789" xx cols).join(" | ");
27
28 my \total-rows = ceiling($len / 10);
29 my \rows = ceiling(total-rows / cols);
30
31 for 0 ..^ rows -> $j {
32     for 0 ..^ cols -> $p {
33         print(" | ") if $p != 0;
34         my $q = $j + rows * $p;
35         next if $q >= total-rows;
36         printf("%3d ", $q);
37         for 0 ..^ 10 {
38             my $i = 10 * $q + $_;
39             if $i >= $len { print " "; }
40             else {
41                 print "0123456789XE".substr(@k[$i], 1);
42             }
43         }
44     }
45     say "";
46 }

```

workfiles/7/bins.pl6

9.8 | workfiles/7/conno.pl6

```

1 my $digits-str = "0123456789TKXSNVFM";
2 my @digits = $digits-str.comb;
3
4 sub convert-small-fwd($n, $pad = False) {
5     die "$n must be < 4199" if $n >= 4199;
6     my $a = $n div (19 * 17);
7     my $b = ($n div 17) % 19;
8     my $c = $n % 17;
9     return
10         (!$pad && $a == 0 ?? "" !! @digits[$a]) ~
11         (!$pad && $b == 0 && $a == 0 ?? "" !! @digits[$b]) ~
12         @digits[$c];
13 }
14
15 sub convert-small-back($s) {
16     die "$s must be 3 chars or fewer" if $s.chars > 3;
17     my $c = $digits-str.index($s.substr(* - 1, 1) // "0");
18     my $b = $digits-str.index($s.substr(* - 2, 1) // "0");
19     my $a = $digits-str.index($s.substr(* - 3, 1) // "0");
20     return $c + 17 * ($b + 19 * $a);
21 }
22
23 sub triangle($n, $p) {
24     return ($n * (2 * $p + 1 - $n)) div 2;
25 }
26
27 sub sqrt-floor($y) {
28     die "$y is negative" if $y < 0;
29     return $y if $y < 2;
30     my $small = sqrt-floor($y +> 2) +< 1;
31     my $large = $small + 1;
32     return $small if $large * $large > $y;
33     return $large;
34 }
35
36 sub sqrt-ceil($y) {
37     my $n = sqrt-floor($y);
38     return $n if $n * $n == $y;
39     return $n + 1;
40 }
41
42 sub untriangle($y, $p) {
43     return (2 * $p + 1 - sqrt-ceil(4 * $p * $p + 4 * $p - 8 * $y + 1))
44         div 2;
45 }
46
47 my @powers = (4199);
48
49 for 0 .. 10 {
50     my $p = @powers[* - 1];
51     @powers.push: $p * ($p + 1) div 2;
52 }
53
54 sub convert-large-fwd-h($n, $i, $pad = False) {

```

```

53 # base case
54 if $i == 0 {
55     return convert-small-fwd($n, $pad);
56 }
57 # recursive
58 my $super = untriangle($n, @powers[$i - 1]);
59 my $infra = $n - triangle($super, @powers[$i - 1]);
60 if $super == 0 && !$pad {
61     return convert-large-fwd-h($infra, $i - 1, False);
62 }
63 return
64     convert-large-fwd-h($super, $i - 1, $pad) ~
65     (":" x $i) ~
66     convert-large-fwd-h($infra, $i - 1, True);
67 }
68
69 sub convert-large-fwd($n, $pad = False) {
70     my $i = 0;
71     ++$i while @powers[$i] <= $n;
72     convert-large-fwd-h($n, $i, $pad);
73 }
74
75 sub convert-large-back($s) {
76     # Find the longest run of colons
77     my @matches = ($s =~ m:g/":+"/); #/"
78     if (!@matches) {
79         return convert-small-back($s);
80     }
81     my $longest-match = @matches.max(*.chars);
82     my $i = (~$longest-match).chars;
83     my $left = $s.substr(0, $longest-match.from);
84     my $right = $s.substr($longest-match.to);
85     my $sup = convert-large-back($left);
86     my $inf = convert-large-back($right);
87     return triangle($sup, @powers[$i - 1]) + $inf;
88 }
89
90 multi MAIN(Int :$fwd) {
91     say convert-large-fwd($fwd);
92 }
93 multi MAIN(Str :$back) {
94     say convert-large-back($back);
95 }

```

workfiles/7/conno.pl6

9.9 | workfiles/7/count-days.pl6

```

1 # Count the number of days between 1/0/0 and D/M/Y, inclusive.
2
3 # CONSTANTS
4
5 constant \MONTHS_PER_YEAR_CYCLE = 7215;
6 constant \YEARS_PER_YEAR_CYCLE = 736;

```

```

7 constant \AVG_MONTHS_PER_YEAR = MONTHS_PER_YEAR_CYCLE /
  YEARS_PER_YEAR_CYCLE;
8 constant \MONTHS_PER_MONTH_CYCLE = 136;
9 constant \DAYS_PER_MONTH_CYCLE = 4053;
10
11 # COMPUTATION
12 # For each year, take as many months as are needed
13 # in order to cycle to the next.
14
15 my $c = 0;
16 my @k = (0);
17
18 for 0 ..^ YEARS_PER_YEAR_CYCLE -> $i {
19   my $need = 1 - ($c - floor($c));
20   my $objs = ceiling($need * AVG_MONTHS_PER_YEAR);
21   @k[$i + 1] = $objs;
22   $c += $objs / AVG_MONTHS_PER_YEAR;
23 }
24
25 my @cumk = [\+] @k;
26
27 sub months-before-year($year) {
28   my $whole-cycles = $year div YEARS_PER_YEAR_CYCLE;
29   my $remainder = $year % YEARS_PER_YEAR_CYCLE;
30   return $whole-cycles * MONTHS_PER_YEAR_CYCLE + @cumk[$remainder];
31 }
32
33 my @m = (0);
34
35 for 0 ..^ MONTHS_PER_MONTH_CYCLE -> $i {
36   @m.push: ($i % 5 == 2) ?? 29 !! 30;
37 }
38
39 my @cumm = [\+] @m;
40
41 sub days-before-month($month) {
42   my $whole-cycles = $month div MONTHS_PER_MONTH_CYCLE;
43   my $remainder = $month % MONTHS_PER_MONTH_CYCLE;
44   return $whole-cycles * DAYS_PER_MONTH_CYCLE + @cumm[$remainder];
45 }
46
47 sub days-before-date($d2, $m, $y) {
48   my $d = $d2 - 1; # d is 0-indexed
49   my $bm = months-before-year($y) + $m;
50   return days-before-month($bm) + $d;
51 }
52
53 sub MAIN($d2, $m, $y) {
54   say days-before-date($d2, $m, $y);
55 }

```

workfiles/7/count-days.pl6

Romanisation

In this text, the romanisation is used only to transcribe names into English. Whenever possible, the hacmisation should be used.

Table 9.4: The consonants of aaaaaaaaA.

	Bilabial	Alveolar	Palatal	Velar	Glottal
Nasal	m	n	ɲ	ŋ	
Plosive	p b	t d	tʃ dʃ	k g	ʔ
Fricative	f	s	ʃ	h	
(coarticulated)	ɸh	fh		fʃ	
Affricate		ts	tʃ		
Lateral fricative		ɬ			
Approximant		r	j	w	
Lateral approximant		l			
Trill		ʀ			

Table 9.5: The vowels of aaaaaaaaA.

Spread	Half-rounded	Rounded
i	y	ɥ
ĩ	u	û
e		ö
ë		o
a		

Rod signs are represented by the Arabic digits <1 2 3 4 5 6 7 8> attached to the end of the verbs they encompass. Proper words are preceded by a backslash <\>.