

---

ᐃᐅᐅ ᐅᐅᐅ ᐅᐅᐅ  
*Jbl, the language of Nᐅᐅ*

---

uruwi

|||||||||||||  
aaaaaaaaaaaaaaaa  
*A complete grammar*

28 May 2018

*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 . . . . .	4
0.1.1	Synopsis . . . . .	4
0.1.2	External history . . . . .	4
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
1.6	Vocalisation . . . . .	7
2	The statement space . . . . .	9
2.1	Conceptualisation . . . . .	9
2.2	Application . . . . .	9
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 . . . . .	13
2.5.2	Quantifiers . . . . .	14
2.6	Pro-forms . . . . .	15
2.6.1	Pro-forms of the zeroeth kind . . . . .	15
2.6.2	Pro-forms of the first kind . . . . .	16
2.6.3	Pro-forms of the second kind . . . . .	16
2.6.4	Pro-forms of the third kind . . . . .	16
2.6.5	Pro-forms of the fourth kind . . . . .	17
2.7	Seasonal cycles . . . . .	17
2.8	Questions . . . . .	17
2.9	Conjunctions . . . . .	18
2.9.1	Appositives . . . . .	19
2.10	Subordinate clauses . . . . .	19
2.11	Connectors . . . . .	19
2.12	Proper words . . . . .	21
2.13	Variables . . . . .	21
2.13.1	Lambdas . . . . .	22

2.13.2	Arrays . . . . .	22
3	Numerals . . . . .	23
3.1	Fractional numerals . . . . .	24
3.2	Mathematical operators . . . . .	24
3.3	Mathematical constants . . . . .	25
3.4	Figures in signage . . . . .	25
4	Word formation . . . . .	27
4.1	Compounding . . . . .	27
4.1.1	Coördinating compounds . . . . .	28
4.1.2	Subordinating compounds . . . . .	28
A	Dictionary . . . . .	29

## 0.1 | Introduction

### 0.1.1 | Synopsis

*Jbl* /jbl/ is the official language of *Njln*, and it is also widely spoken across the world. Unusually, it allows speakers to interchange consonants with their corresponding vowels at will. In other words, the name of the language can be pronounced as [jyl], [ipw], [iyl] or so on. (The author prefers to pronounce it [jyl].) Because text is written with only the consonant glyphs, it carries the illusion that the language has no vowels. Airflow direction is also phonemic.

Grammatically, *Jbl* is an analytic SVO or VSO language, but the distinction between nouns and verbs is less clear than another line of division: that between *concretes* and *abstracts*. It marks concretes for five noun classes, as well as *number mutability*, which indicates how likely the quantity of an item is to change.

*Jbl* is also well-known for allowing speakers to use any numerical base they please, even allowing them to change it on the fly, as well as having a number of nonconservative determiners.

### 0.1.2 | External history

Work on *Jbl* began on 12 November 2017 (ᐃᐱ ᓂᓂ ᐱᐱ), when it was tentatively known as *leucᐢᐱ* (literally *8th conlang*), and the files were committed to the Git repository two days later. The official name of the language was decided on 20 December 2017 (ᐃᐱ ᐃᐢ ᐃᐢᐢ). Fun fact: this is the only paragraph where hacm digits will appear! All this, despite the fact that *Jbl* uses hacm!

The official name of the language was decided when Uruwi called it the “yellow paper language”, after the colour of the paper used for it. Mareck suggested abbreviating it to “ypl” (which would be transcribed <ɥdl>), but since there is no <d> in the language, it was replaced with a <b>.

# 1 | Phonology and orthography

## 1.1 | Phoneme inventory

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

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

Table 1.1: Phonemes of Jbl.

#	Hacm	Roman	Consonant (Egre) (Ingr)		Vowel
0	ʃ	t	tʰ	ʔ	eɪ
1	nʰ	ŋ	ŋ	ɡ	ā
2	b	b	p	p	y
3	ɒ	m	m	ɓ	ũ
4	j	s	s	s	i
5	ɥ	j	j	j	i
6	l	ç	tʰ		ʌɪ
7	o	w	w	w	u
8	ɸ	g	k	k	o
9	l	l	l	l	uu
10	ɫ	k	kʰ	ʈ	oɪ
11	n	n	n	n	ẽ
12	ɾ	ř	r	ɹ	ẽ
13	ɒ	d	t	t	e
14	h	h	h	h	a
15	s	ž	ʈ	ʈ	ʌ

When pronounced ingressively, the tones of vowels are inverted. That is, [ʌɪ] becomes [ʌɪ̯]. 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  $\langle / \rangle$  (to ingressive) and  $\langle \backslash \rangle$  (to egressive). In dictionaries, a switch in airflow direction (without regard to the final state) is marked using  $\langle \rangle$ .

### 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 or interleaving, then the violation is resolved by:

- replacing the earlier instance with an instance of its predecessor (e. g.  $/w/$  (7)  $\rightarrow /t\ddot{t}'/$  (6), wrapping when necessary), and
- replacing the later instance with an instance of its successor (e. g.  $/w/$  (7)  $\rightarrow /k/$  (8), wrapping when necessary).

### 1.4 | Allophony

The following changes are made:

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

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

Thus, for instance,  $/hswlm\eta/$  would be resolved to  $[hswp\eta]$ , which could, for instance, be pronounced  $[asup\grave{a}]$ .

### 1.5 | The biting affix

A frequent type of affix encountered in Jbl is the *biting affix*, which has the syntax  $\langle \rightarrow \delta \omega \rangle$ . To apply this affix onto a word  $\alpha$ :

- Take the last  $\text{length}(\delta)$  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  $\langle \text{dnl} \rangle$ , then we would:

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

## 1.6 | Vocalisation

Vocalisation is the process of determining which phonemes to pronounce as vowels, and which as consonants. In usual speech, speakers tend to use convenient pronunciations; for instance, ingressive nasal vowels are generally avoided, as are complex consonant or vowel clusters. A speaker who is unable to pronounce  $[\text{r}]$  will usually set things up so any egressive  $\langle \text{r} \rangle$ s will be pronounced as vowels instead of consonants. Nonetheless, formal vocalisation patterns are used more often in chants or songs.

- Vowel-only ( $\langle \text{jlf} \rangle$ ): in strong forms of this vocalisation pattern, all phonemes are pronounced as vowels. Weaker versions of this pattern might pronounce ingressive nasal phonemes as consonants.
- CV ( $\langle \text{nb} \text{r} \text{n} \text{f} \text{l} \rangle$ ): alternates between consonants and vowels, starting with a consonant. Words with odd numbers of phonemes will have the last one pronounced as a consonant. Words with one phoneme are pronounced as vowels.
- VC ( $\langle \text{jf} \text{b} \text{l} \rangle$ ): alternates between vowels and consonants, starting with a vowel. Words with odd numbers of phonemes will have the last one pronounced as an extra consonant. Words with one phoneme are pronounced as vowels.

Thus, here are some vocalisations for a sentence:

$\backslash \varphi \text{ d} \varphi \text{j} / \text{n}^{\text{e}} \text{ns} \text{] jn}^{\text{e}} \backslash \varphi \text{ l} / \text{b} \text{l} \text{a} \text{ } \varphi \backslash \text{n}^{\text{e}} \text{p} \text{h} \text{ j} \varphi \text{d} \text{ j} \text{a} / \text{b} \text{d} \varphi \backslash \text{l} \text{f} \text{n} \text{ n} \varphi / \text{f} \text{l} \text{a} \text{ } \backslash \text{r} \text{f} \text{d} / \text{d} \text{l} \text{.}$

$/ \uparrow \text{k} \text{ mks} \downarrow \text{njtjk}' \text{ s} \text{nj} \uparrow \text{t}' \text{kl} \downarrow \text{k}' \text{plt} \text{ kl} \uparrow \text{njmh} \text{ skm} \text{ st} \downarrow \text{pmj} \uparrow \text{t}' \text{t}' \text{n} \text{ nj} \downarrow \text{k}' \text{lt} \uparrow \text{rk}' \text{m} \downarrow \text{mt}' \text{k}' /$

Common (one possibility):

$[ \uparrow \text{o} \text{ mos} \downarrow \text{gei} \uparrow \text{ig}' \uparrow \text{t}' \text{ol} \downarrow \text{tule} \text{ ol} \uparrow \text{nj} \text{a} \text{ som} \text{ se} \downarrow \text{ymi} \uparrow \text{t}' \text{e} \text{ln} \text{ ni} \downarrow \text{tut} \uparrow \text{ro} \text{lm} \downarrow \text{m} \text{a} \text{J} ]$

/r/-less (one possibility):

$[ \uparrow \text{o} \text{ mos} \downarrow \text{gei} \uparrow \text{ig}' \uparrow \text{t}' \text{ol} \downarrow \text{tule} \text{ ol} \uparrow \text{nj} \text{a} \text{ som} \text{ se} \downarrow \text{ymi} \uparrow \text{t}' \text{e} \text{ln} \text{ ni} \downarrow \text{tut} \uparrow \text{ek}' \text{u} \downarrow \text{m} \text{a} \text{J} ]$

Vowel-only:

$[ \uparrow \text{o} \text{ } \text{uoi} \downarrow \text{a} \text{eio} \text{] i} \text{a} \uparrow \text{e} \text{lou} \downarrow \text{o} \text{yue} \text{ ou} \uparrow \text{a} \text{u} \text{a} \text{ io} \text{u} \text{ ie} \downarrow \text{yui} \uparrow \text{a} \text{le} \text{le} \text{e} \text{i} \downarrow \text{o} \text{lue} \uparrow \text{go} \text{l} \text{u} \downarrow \text{u} \text{a} \text{lo} \text{J} ]$

CV:

$[ \uparrow \text{o} \text{ mos} \downarrow \text{ati} \text{s} \text{a} \uparrow \text{t}' \text{ol} \downarrow \text{tule} \text{ ku} \uparrow \text{nj} \text{h} \text{ som} \text{ se} \downarrow \text{p} \text{uj} \uparrow \text{a} \text{t}' \text{e} \text{ni} \downarrow \text{tut} \uparrow \text{ro} \text{lm} \downarrow \text{m} \text{a} \text{J} ]$

VC:

$[ \uparrow \text{o} \text{ } \text{uki} \downarrow \text{gej} \uparrow \text{ig}' \uparrow \text{e} \text{kl} \downarrow \text{o} \text{puut} \text{ ol} \uparrow \text{am} \text{h} \text{ ikm} \text{ it} \downarrow \text{ymi} \uparrow \text{t}' \text{e} \text{ln} \text{ ej} \downarrow \text{o} \text{lt} \uparrow \text{ek}' \text{m} \downarrow \text{u} \text{J} ]$





## 2 | The statement space

### 2.1 | Conceptualisation

*Synopsis:* *jbl* 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

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

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

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

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:

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

At the end of the sentence, any number of  $\lrcorner$ s can be omitted.

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

Table 2.2: Names of syntactic markers in Jbl.

$\triangle$	js
$\blacktriangleright$	h
$\ulcorner$	$\rightarrow$ q:fn
$\lrcorner$	b
$\lrcorner$	$\varphi$
$\lrcorner$	$\rightarrow$ 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 Jbl.

(° 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

(° 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
$\Omega$	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.
$\vec{\Omega}$	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.
$\Sigma$	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	->n <sup>0</sup> :	Ω	lbD
2	->j:	Ψ	lΩn <sup>0</sup>
3	->l:	X	bΩj
4	->b:	Z	l <sub>l</sub> q
5	->n:	Ξ	DyD
0	->h:	Φ	ΩSl
		Ω̇	lbq
		Ψ̇	lhD
		Ẋ	bhl
		Ξ̇	D <sub>l</sub> q
		Φ̇	Ω <sub>l</sub> d
		Σ	ln <sup>0</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:

- <boφ> c2 (0) eats (1)
- <φnl/ > 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:

\φnl/jDyD bo\rdyD Ωs/lDyD  
 φnlf->j:DyD boφ->j:DyD Ωsn->n<sup>0</sup>:DyD  
 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:

\φnl/jDyD boφ Ωsn  
 φnlf->j:DyD 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:

\\p \\p fo/lbnj d\\j\\n<sup>o</sup>ns/lbh \\p lb\\j\\lbnj j/nbh \\p\\n<sup>o</sup>dh  
 \\p \\p \\p|→n<sup>o</sup>:bnj d\\js→h:ns|→d:bh \\p lb|n→j:bnj jn<sup>o</sup>→d:bh \\p|h→d:bh  
 ⊥ ⊥ sun-1- $\vec{X}$  weather-0- $\Phi$ - $\neg$  ⊥ summer\_solstice-2- $\vec{X}$  after- $\neg$  morning( $\vec{X}$ )- $\neg$

The rest of the sentence is thus:

j\\p d jn/bdy\\l\\n lln ns/n<sup>o</sup>n d\\p\\p\\p b dl  
 j\\p d jn→n<sup>o</sup>:dy→y:ln lln ny→y:ln d\\d→n:dy b dl  
 time walk-1- $\Xi$ - $\neg$  speaker( $\Xi$ ) destination- $\neg$  mountain-5- $\Xi$ - $\neg$  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:

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

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

$\backslash \phi$   $D\phi j/n^\phi ns$   $j n^\phi$   $\backslash \phi |$   $/b l a$   $\phi | \backslash n^\phi D h$   
 $\phi$   $D\phi j s \rightarrow h: ns$   $j n^\phi$   $> \phi |$   $> b l a$   $\phi | h \rightarrow r: b h$   
 $\perp$  weather-0- $\phi$  after D-sun D-summer\_solstice morning- $\neg$

$j \phi \rangle$   $j n^\phi /b d y \backslash l r n$   $n y$   $/ l a$   $\backslash r l a$   $/ d l$   
 $j \phi \rangle$   $j n^\phi \rightarrow n^\phi: D y p \rightarrow y: r n$   $n y$   $> l a$   $> r l a$   $> d l$   
time walk-1- $\Xi$ - $\neg$  destination D-speaker D-mountain D-start

Other descriptors are possible:

Table 2.7: Other descriptors.

Prefix	Input	Output
$> f'$	$C$	Inalienable_Possession( $C$ )
$> f \Omega'$	$C$	Association( $C$ )
$> \phi b'$	$C$	Property( $C$ )
$> l l'$	$C$	Borrow( $C$ )
$> j h'$	$C$	Destination( $C$ )

### 2.5.2 | Quantifiers

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

$$\text{All } Xs \text{ are } Ys \equiv X \subseteq Y \quad (2.7)$$

$$\text{Some } Xs \text{ are } Ys \equiv (X \cap Y) \neq \emptyset \quad (2.8)$$

$$\text{No } Xs \text{ are } Ys \equiv (X \cap Y) = \emptyset \quad (2.9)$$

Table 2.8 lists the *conservative* quantifiers of Jbl. 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 ( $Q(X, Y)$ )
ḥy	All	$X \subseteq Y$
yn <sup>φ</sup>	Not all	$X \not\subseteq Y$
ny <sup>φ</sup>	Some	$(X \cap Y) \neq \emptyset$
bo	None	$(X \cap Y) = \emptyset$
dyo	Most	$ X \cap Y  \geq  X - Y $
phi	At least two	$ X \cap Y  \geq 2$
on <sup>φ</sup>	One	$ X \cap Y  = 1$
onn <sup>φ</sup>	Half of	$  X \cap Y  -  X - Y   \leq 1$
bl	A finite number of (sometimes “many”)	$ X \cap Y  < \aleph_0$
sjh	A countable number of	$ X \cap Y  \leq \aleph_0$
nl	An infinite number of	$ X \cap Y  \geq \aleph_0$
fn <sup>φ</sup>	An uncountable number of	$ X \cap Y  > \aleph_0$
lj	Almost all	$ X - Y  < \aleph_0 \wedge  Y  \geq \aleph_0$ $ X - Y  \leq \aleph_0 \wedge  Y  > \aleph_0$

Table 2.9: Nonconservative quantifiers.

Quantifier (Q)	Meaning of Q Xes are Y ( $Q(X, Y)$ )
o <sup>φ</sup> b	$Y \subseteq X$
yno <sup>φ</sup>	$Y \not\subseteq X$
phi	$ X  =  Y $
dj <sup>φ</sup>	$ X  <  Y $
sj <sup>φ</sup>	$ X \cap Y  \geq  Y - X $

- 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:

- <ln> the speaker or writer
- <dnj> 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  $\langle -\alpha \rangle$ :  $\langle \phi n l \rangle \rightarrow \langle \phi n \alpha \rangle$ . These pro-forms can be described 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
sfo	self	
ysn <sup>o</sup>	adversary	
rdhy	friend, ally	
φjα	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,  $\langle obl\alpha \rangle$  refers to a teacher of the previously-referred concrete, and  $\langle \alpha obl \rangle$  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:

- $\langle s \rangle$  refers to a temporal abstract (not necessarily the one that was mentioned the latest)
- $\langle no \rangle$  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  $\langle \text{յօժ յո՞ր ըլ ըլ լ փոհ} \rangle$ .

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

### 2.6.5 | Pro-forms of the fourth kind

The phrase  $\langle \text{հոլլ պփօ} \rangle$  is an abstract that refers to a sentence previously said by the listener, and it is usually used alone. It can also modify a sentence that disambiguates what was referred to.

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

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

## 2.7 | Seasonal cycles

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

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

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

Perhaps the most well-known quintuplet is  $\langle \langle \text{յփ, ռՏյ, Ծճո՞, հլլ, օրհ} \rangle \rangle$  which, in season 1, correspond to the five seasons in order. Thus, to refer to the second season, one would say:

- $\langle \text{ռՏյ} \rangle$  during the first season
- $\langle \text{հլլ} \rangle$  during the fourth season

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

## 2.8 | Questions

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

This replaces the top-level  $\blacktriangleright$ , 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:

Table 2.12: Interrogative words in Jbl.

Particle	Marker	Definition
hrlj	rljh	what, who
fob	obf	when, where
sln	lns	why, how
φyn	ynφ	how many, how much
lhñ	hñl	to what extent

\hrlj bo/rlbð rljh ðñ\ðyð  
 hrlj boφ-→j:lðð rljh ðñl-→l:ðyð  
 P-what eat-2-Ω Q-what bread-3-Ξ  
 Who ate the bread?

\hrlj ll/ðlðð  
 hrlj llñ-→h:lðð  
 P-what speaker-0-Ω  
 Me.

However, <sln-lns> questions are usually answered with full sentences.

<lhñ> and <hñl> are special – they request a number, usually between 0 and 1, inclusive<sup>1</sup>, that states to what degree the hypothesis is true. In that sense, they can be used to ask polar questions.

## 2.9 | Conjunctions

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

Some common conjunctions are:

- <ns> and (for items), or (for predicates)
- <slj> or (for items), and (for predicates)
- <nsnj> xor (for both items and predicates)

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

\φ ðhφð rl/φðly \φ/φbhl n\ðbh rlð/yðly  
 φ ðhφð rlð-→n:ðly >fφl-→n<sup>φ</sup>:bhl ns-→rl:bh rlðh-→n:ðly  
 ⌊ below mountain-5-Ξ DESC-sun-1-Ξ and-⌈ valley-5-Ξ

hñf lðfñ n\lñ hñ<sup>φ</sup> b/rlhs\ lh<sub>y</sub>  
 hñf lðfñ ns-→y:lñ hñ<sup>φ</sup> b/ð-→h:ns\ lh<sub>y</sub>  
 create hope and-⌈ life person-0-Φ all

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

<sup>1</sup>A value less than 0 or greater than 1 can be interpreted as an emphatic answer.

Conjunctions are evaluated with left associativity but can be grouped using  $\langle \alpha \varphi \alpha \dots \alpha \alpha \rangle$ .

### 2.9.1 | Appositives

The conjunction  $\langle \alpha \rangle$  = joins the two nodes around it and returns a single node that equates its arguments and refers to the entity in question.

$\backslash \backslash / \text{ speaker } 0 \text{ } \Xi \text{ walk village } 3 \text{ } \Omega \text{ } \neg \text{ association } \neg \text{ uncle } \sqcup \text{ association.DESC-speaker } \sqcup =$

$\text{ village } 3 \text{ } \vec{X} \text{ } \neg \text{ large DUMMY DESC-superlative}$

*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  $\langle \alpha \rangle$  (if there is anything afterward):

$\backslash \backslash / \text{ speaker } 0 \text{ } \Xi \text{ walk again SUB-where see speaker rdkbe } 5 \text{ } 1 \text{ } \Xi \text{ existence DUMMY DESC-Q-where}$

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

$\backslash \backslash \text{ young\_person } 1 \text{ } \Xi \text{ finite = SUB-who existence wreath head DESC-flower borrow.DESC-Q.who } \sqcup$

$\text{ dance environs DUMMY DESC-fire}$

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

## 2.11 | Connectors

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

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

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

Connectors  $x$  and  $y$  are part of the same set  $S$  iff all of the following conditions hold:

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

For the purposes of positioning connectors, two consecutive instances of  $\langle b \rangle$  within the same sentence is treated as one word.

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

Table 2.13: Connectors.

Name	Arity	Jbl	Explanation
Sequential	$n$	יְיִף	A sequence of events. An event $\alpha$ is said to happen before $\beta$ if the sentence describing $\alpha$ is uttered before that describing $\beta$ .
Parallel	$n$	יְיִף	Two or more events happening simultaneously.
Analogous	2	יְיִף	“For the same reason $\alpha$ is true, $\beta$ is also true.” Also used as an “and” without stating any order.
Subversive	$n$	יְיִף	“ $\theta_1$ but $\theta_2$ but $\theta_3$ but etc.”
Augmentative	$n$	יְיִף	Later statements apply to a greater extent than earlier statements.
Explanatory	$n$	יְיִף	“ $\theta_1$ causes $\theta_2$ causes $\theta_3$ etc.”
Conditional	2	יְיִף	“If $\alpha$ , then $\beta$ .”

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

\פול/דפד בִּפּוּ יְיִף אִשָּׁה.  
 פול->דפד בִּפּוּ יְיִף אִשָּׁה  
 fish-2- $\exists$  eat SEQUENTIAL flower  
 The fish ate the flower.

\דוד/פולס דבד יְיִף אִשָּׁה.  
 דוד->נפ:לס דבד יְיִף אִשָּׁה >דוד  
 young\_person-1- $\Omega$  dance SEQUENTIAL surroundings DESC-tree  
 Then the child danced around the tree.

\דוד בִּפּוּ יְיִף פִּי.  
 דוד בִּפּוּ יְיִף פִּי  
 young\_person#PF1 eat SEQUENTIAL fish#PF1

Then the child ate the fish.

Note that <uj> is the third word of all of the sentences above.

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

## 2.12 | Proper words

Because many foreign languages use only egressive airflow, all proper words in Jbl switch to egressive at the beginning.

Sometimes, the status of a word as proper might be ambiguous without hints. Optionally, the prefix <ɸɸ-> can be prefixed (*before the airflow change*) to show that this should be treated as a proper word.

Note that <ɸbl> *jbl* is not a proper word.

## 2.13 | Variables

**Synopsis:** *Variables* provide a more explicit alternative to pro-forms.

Variables are declared to be of a type (implicitly). A type is one of the following:

- a concrete
- an abstract
- a numeral (of which Boolean values are a subset)
- an array consisting of elements of the same type
- a function receiving zero or more parameters and returning a value

Concrete variables are divided into generic and specialised variables:

- Generic variables can refer to different objects depending on how they are used (e. g. *X* in *if there are at least six Xes in the pile, then take one X from your pocket and throw it in*).
- Specialised variables refer to the same entity every time they are used (e. g. *Y* in *if Y is here, then give Y this paper*).

Generic variables are inflected for mutability, but they are always of the first gender.

A variable can be referred to by an arbitrary string, although in practice, generic variables have a minimum identifier length of three characters. Orthographically, they receive the sigil <Ψ>.

Variables can be bound by using *lambdas*.

### 2.13.1 | Lambdas

The syntax for a lambda is

$$\langle \text{f} \text{b} \text{l} \text{b} \rangle ((v \langle n \rangle)^* v)? \langle \text{J} \rangle \text{body} \langle \text{J} \rangle \quad (2.10)$$

where  $v$  is an identifier. A function can subsequently be called with the following syntax:

$$f \langle \text{b} \text{l} \text{b} \rangle ((x \langle n \rangle)^* x)? \langle \varphi \text{J} \rangle \quad (2.11)$$

Note the analogy with the lambda syntax in the argument list.  $f$  need not be bound to a variable; in fact, the lack of such a requirement is the only way to bind variables!

### 2.13.2 | Arrays

The syntax for an array is

$$\langle \text{d} \text{f} \text{y} \text{o} \rangle ((x \langle n \rangle)^* x)? \langle \text{o} \text{y} \text{f} \text{d} \rangle \quad (2.12)$$

and an array is indexed as such:

$$\langle \text{d} \text{f} \rangle A \langle \text{J} \varphi \rangle n \langle \varphi \text{J} \rangle \quad (2.13)$$

As with lambdas, arrays do not need to be bound to a variable to be used.

\varphi y n d r / l b d n b l \varphi y n \varphi o / n h \varphi d  
 \varphi y n d r j - > h : l b d n b l j - > n \varphi : l n \varphi o > n h \varphi d  
 QP.NUM PR0.LISTENER-0-Ω want piece-1-Σ DESC-sugar  
 d n d f y o s n \varphi n s y o y f d j \varphi y n \varphi \varphi j  
 INDEX( ARRAY( two , three ) ARRAY INDEX, QW.NUM ) INDEX  
 What is the number  $i$  for which you want  $[2, 3][i]$  pieces of sugar?  
 or more idiomatically: Do you want two or three pieces of sugar?

### 3 | Numerals

**Synopsis:** *Jbl 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	nd	┐	16	sj
┌	1	sf	┐	17	rlj
└	2	sn <sup>φ</sup>	┐	18	sn
┐	3	sy	┐	19	rln
┐	4	sb	┐	20	rl
┐	5	rl	┐	21	rln <sup>φ</sup>
┐	6	sl	┐	22	nl
┐	7	ro	┐	23	ny
┐	8	sd	┐	24	sφ
┐	9	sl	┐	25	rlh
┐	10	rlφ	┐	26	nh
┐	11	nφ	┐	27	sh
┐	12	so	┐	28	nn
┐	13	no	┐	29	nh
┐	14	rl	┐	30	rb
┐	15	rlf	┐	31	nf

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,  $\langle \backslash \text{rl} / \text{sj} \backslash \text{syso} \rangle$  represents  $C35_{16} = 3125_{10}$ . This number can also be written:

$\backslash \text{nd} / \text{rh} \backslash \text{ndrl}$   
 $\text{zt} \text{†}$   
 $50_{\text{p}0}$

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

An instance of  $\langle \text{rd} \rangle$  followed by a digit  $d$  is equivalent to a series of  $d$  zeroes, so the same number can be written  $\langle \backslash \text{nd} / \text{rl} \backslash \text{rdsbsf} \rangle$ .

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  $\langle \circ \rangle$ , read  $\langle \text{oj} \rangle$ . 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:

$\backslash \text{slq} / \text{sj} \backslash \text{sbsbrdorlndrl}$   
 $\text{jo} \text{†} \text{r} \text{r} \text{r} \text{r}$   
 $6. \text{c} 487ED5$

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 | Mathematical operators

Shown in 3.2. Because of separate abstract forms of operators, it is rather uncommon to group them using  $\langle \text{nyq} \dots \text{non}^{\text{p}} \rangle$ .

Note that  $\langle \text{dlrd} \rangle$  can be used to represent rational numbers.



Table 3.2: Mathematical operators in Jbl.

Conjunction	Abstract	Definition
ḃḵḶ	ḶḶḃ	$a + b$
ḶḃḶ	ḶḶḃ	$a - b$
ḵḶḵḶ	ḶḶḵḶ	$a \cdot b$
ḶḶḶ	ḶḶḶ	$a/b$
ḶḶḶḶ	ḶḶḶḶ	$a^b$
ḶḶḶḶ	ḶḶḶḶ	$a + bi$

3.3 | Mathematical constants

Some numbers have their own names (shown in 3.3).

Table 3.3: Mathematical constants in Jbl.

Abstract	Definition
ḶḶḶḶ	$1/2$ (identical to $\langle ḶḶḶḶ \rangle$ )
ḶḶḶḶ	$2 \cdot \pi$
ḶḶḶḶ	$e$
ḶḶḶḶ	$\sqrt{2}$
ḶḶḶḶ	$\sqrt{3}$
ḶḶḶḶ	$i = \sqrt{-1}$
ḶḶḶḶ	$\zeta(3)$

3.4 | 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,  $\langle Ḷ \rangle$  is displayed as  $\langle ḶḶḶ \rangle$ . This happens even in contexts in which fractional numbers do not make sense.

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



## 4 | Word formation

### 4.1 | Compounding

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

Jbl 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.

(Another way to describe this division is that, in terms of more conventional terminology, *coördinating* compounds usually cover copulative and appositional compounds, and *subordinating* compounds usually cover endocentric and exocentric 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  $\langle \phi lh, \alpha sn, \iota \phi l \rangle$  can be interleaved into  $\langle \phi \alpha \iota l s \phi h n l \rangle$ .

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 constituents, if the longest word is not also the first, then its index is prefixed:  $\langle \jmath \alpha \phi, \phi lh, \phi l r \phi \rangle$  makes  $\langle s n \phi \jmath \phi l \alpha l r \phi \phi \rangle$ .

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:  $\langle \phi h \phi, \phi n \phi \phi \rangle$  makes  $\langle s \phi \phi \phi h n h r \phi \phi \phi \rangle$ .

#### 4.1.1 | Coördinating compounds

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

#### 4.1.2 | Subordinating compounds

Unlike in coördinating compounds, the constituents of a subordinating compound is order-dependent. In particular, the constituents are put in head-initial order with right-associativity. For instance, <φlh, Ωsn, fφl> *morning, flower, sun* means *morning of sunflowers* (not *flower-morning of the sun*).

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

# A | Dictionary

An entry looks like this:

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

From left to right:

1. The entry – the Jbl 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.

1 (

(l $\phi$ ,  $\Omega$ S,  $\Delta$ W, h $\eta$ , old) qcccc  
the five seasons (LSp-ESu, LSu-EAu, MAU-  
EWi, MWinter, LWi-ESp)

(ɔjɸ, jhɔ, sɪb, ɔnl, hnʋf) qcccaa  
(asparagus, cucumber, carrots, percep-  
tion, life)

(jsb, ɥɥ, jɲ, bɸl, ɳbɳ) qcccc  
(rose, dandelion, kosmos, beetroot, elder-  
berry)

(փոռ, եկ, ձի, ռ՞օռ, սոճ) *qc-cccc* (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.

1 r

ɪnʱɔ uncle, aunt  
 ɪn a finish (modifying) termin-  
 ative, perfect

ḥḥ c1 sun

## | b

bḥḥ c0 person, entity  
boḥ c2 food (0) eats (1)  
bḥ c1 piece  
bḥḥ a size, large  
bhḥ c3 fire

## | D

Dḥḥḥ a equality, (0) equals (1)  
Dḥ a start (modifying) inceptive,  
inchoative  
Dḥ c2 tree  
Dḥḥ c0 weather  
Dḥḥ c2 dance  
Dḥ a loan (1) is borrowing (0)  
[from (2)] (modifying) (0)'s (\*) (bor-  
rowed)  
Dḥ c1 wreath  
Dḥ c0 the listener or reader, you  
Dḥḥḥ c5 head  
Dḥ c3 bread  
Dḥ a edge, boundary, extremity,  
superlative (of (0))  
Dḥḥ a below, underside (modi-  
fying) (\*) is under, below (0); downward  
Dḥḥ a again, repeat

## | j

jḥḥ c3 VC vocalisation  
jḥḥ a after  
jḥ c1 wind, vowel-only vocalisation  
jḥḥ a time when (0), (1) (modi-  
fying) (\*) happened at the time (0)  
jḥ c1 walk (toward (1))

## | ḥ

ḥḥ a the ḥḥ language  
ḥḥ c3 village, town

## | φ

φḥḥ c4 dark, dim  
φḥ c0 morning  
φḥḥ c2 fish  
φḥḥ a absence it is not true that  
(0)

## | l

lḥḥ a hope  
lḥ a desire (0) wants (1) (modi-  
fying) (0) is wanted by someone

## | ḥ

ḥḥ c1 summer solstice  
ḥḥ a environs, (0) surrounds (1)  
(modifying) (\*) surrounds (0), around (0)  
ḥḥ c0 the speaker or writer, I  
ḥḥ c1 young person  
ḥḥ a existence, (0) is at (1) (modi-  
fying) (\*) is at (0)

## | n

nḥḥḥ c2 CV vocalisation  
nḥḥ a destination (0) intends to go  
to (1) (modifying) to (0)  
nḥḥ a sight (0) sees (1)  
nḥḥḥ c3 sugar

## | ḥ

ḥḥ c5 mountain  
ḥḥ c5 valley, crease  
ḥḥ a before

## | ḥ

ḥḥḥ a association, relationship  
(0) is related to (1) [by (2)], (0) is (1)'s  
(2) (by association) (modifying) (0)'s  
(\*) (associated)  
ḥḥ a language  
ḥḥ c1 flower

| h

(owned)

h**φ**b *a* property, belongings (1) is h**η**f *a* creation (0) creates (1)  
 the owner of (0) (modifying) (0)'s (\*) (modifying) (\*) created by (0)