Complexlang: a COMPact Logical EXperimental LANGuage

Kelvin M. Liu-Huang
Carnegie Mellon University
ecesare@gmail.com

Published in *Proceedings of SIGBOVIK 2019* (15 March 2019)
Informal revisions (27 March 2019, 29 April 2019)

Abstract

Natural languages trade compactness and consistency for efficiency. Complexlang is an *a priori*, *declarative*, *ideographic* spoken/written language which attempts to construct/ground the semantic structure of both morphology and syntax from first principles using tools provided by propositional logic, set theory, type theory, number theory, object-oriented programming, metaphysics, linguistics, and classical field theory. In doing so, we hypothesize that speakers may converse in Complexlang with little training and learn some math and science in the process.

1. Introduction

Most previous constructed languages intended for human use set out to improve etymological integrity (Zamenhof, 1887), semantic clarity (Bliss, 1965; Karasev, 2006; Weilgart, 1979; Quijada, 2004), consistency (Weilgart, 1979; Cowan, 1997; Quijada, 2004), or other academic merits. Not many (Weilgart, 1979; Cowan, 1997; Bourland & Johnston, 1991; Quijada, 2004; Lang, 2014) have addressed cognitive benefits. First, the arbitrary phonetics and morphology of most natural languages creates cognitive dissonance, which can be easily averted. Also consider how mathematical expressions can precisely express a great deal using a very small number of definitions. Compared to the ambiguity and learning barrier of natural languages, mathematical expressions seem better in these ways. The tradeoff is, of course, that mathematical descriptions can be very elaborate or unwieldy.

We attempt to address all these concerns by constructing Complexlang. Language should ideally synchronize speech, writing, and comprehension in order to facilitate learning. Like aUI (Weilgart, 1979) and Arahau (Karasev, 2006), by infusing individual letters with meaning and using phonemic orthography, words have transparent and largely deterministic etymology; writing, speech, and meaning can all be inferred from each other, reducing ambiguity, speeding up learning, and even allowing efficient and deterministic creation of neologisms. For simplicity, the orthography is simply the IPA symbols of the phonemes. Unlike aUI and Arahau, Complexlang attempts to express semantics entirely through logic, specifically patterned after set theory, rather than metaphor, resulting in compact, transparent, and unambiguous expressions. Semantically, consonants represent the set of all objects of a certain subtype. Compared to mathematical expressions, introducing these axiomatic sets, instead of rigorously defining everyday objects that people agree on, greatly increases the efficiency of communication. Vowels represent Boolean, set, scalar, vector algebraic functions (possibly multiple all at once because many operations have analogous operations for

Boolean/set/scalar/vector subtypes, and the operator is overloaded). Expressions are formed by selecting subsets containing the desired objects. Morphology is derived from inorder traversal because it is impossible to pronounce preorder and postorder of many trees due to consonant duplicates (vowel clusters also pose a problem).

Not only do we shape language, the Sapir-Whorf hypothesis suggests that language also influences (and perhaps determines) our thoughts and behavior. Some previous conlangs (Weilgart, 1979; Cowan, 1997; Bourland & Johnston, 1991; Quijada, 2004) have attempted to explore or utilize this hypothesis to improve cognitive function, but most achieve this by through increased complexity (Quijada, 2004). Meanwhile, Lojban is largely grounded in logic, though word formation is still arbitrary because they are synthesized from existing languages (Cowan, 1997), contributing to the aforementioned cognitive dissonance. Complexlang attempts to ground both morphology and syntax in logic using tools provided by propositional logic, set theory, type theory, number theory, object-oriented programming, metaphysics, linguistics, and classical

Table 1. All Complexlang vowels, their phonemes (indicated with IPA), and the function they perform (if multiple available, the function is chosen based on the input type).

IPA	Boolean input	set input	quantity input	function input
i		$\triangleleft \overset{\text{def}}{=}$ var *x* in most recent ancestral : or ¤ $\triangleleft(S) \overset{\text{def}}{=}$ previous var of subtype *S* $\triangleleft(S,I) \overset{\text{def}}{=}$ *I*th previous var of subtype *S*	$\triangleleft(I) \overset{\text{def}}{=}$ *I*th previous letter	
a	$\wedge(P,Q,\dots)$	$\cap(S,T,\dots)$	$\times([-,]I,[-,]J,\dots)$ $\overset{\text{def}}{=} I^{[-1]}J^{[-1]}\dots$	
u		$\# \overset{\text{def}}{=} j(.)$ $\#(S) \overset{\text{def}}{=} j(S)$ $\#(S,I) \overset{\text{def}}{=} \{x \in S \mid j(x) = I\}$		
o		$:(S,P) \overset{\text{def}}{=} \cup\{x \in Z(S) \mid P\}$ $:(S,P,X) \overset{\text{def}}{=} \cup\{X \mid x \in Z(S) \wedge P\}$		
e		¤$(S) \overset{\text{def}}{=} deal(Z(S))$ ¤$(S,X) \overset{\text{def}}{=} deal(\{X \mid x \in Z(S)\})$		¤$(O,S) \overset{\text{def}}{=} O\left(deal(Z(S))\right)$ ¤(O,S) $\overset{\text{def}}{=} O(deal(\{X \mid x \in Z(S)\}))$
ɛ	$\rightarrow(P,Q)$	$\subset(S,T)$	$<(I,J)$	
ɔ	$\leftarrow(P,Q)$	$\supset(S,T)$	$>(I,J)$	
ə	$\neg(P)$	$\complement(S)$	$-(I)$	
ɨ		$\boxdot(S) \overset{\text{def}}{=}$ force field created by *S*		
ɪ	$\leftrightarrow(P,Q,\dots)$	$=(S,T,\dots)$	$=(I,J,\dots)$	
ʊ	$\oplus(P,Q,\dots)$	$\Delta(S,T,\dots)$	$\neq(I,J,\dots)$	
y		$\boxplus(S) \overset{\text{def}}{=} \mu(J(S))$	$\boxplus(I) \overset{\text{def}}{=} \lvert I \rvert$	
ɯ			$\delta(I,J) \overset{\text{def}}{=} \frac{dI}{dJ}$	
ɑ	$\vee(P,Q,\dots)$	$\cup(S,T,\dots)$	$+([-,]I,[-,]J,\dots)$ $\overset{\text{def}}{=} [-]I + [-]J + \cdots$	

where $Z(S) = \{\{x \mid x \in S, j(x) = i\} \mid i \in J(S)\}$, $J(S) \subseteq \mathbb{K}^{\mathrm{n}}$ is the index set of *S* using the recommended units, $j: S \rightarrow J(S)$ is the index of the input element in *S*, and $k: S \times J(S) \rightarrow 2^S$ is the subset of *S* with input index

Table 2. All Complexlang consonants, their phonemes (indicated with IPA), and the recommended set they represent.

IPA	emoji	set subtype, $t(S)$	recommended elements in set, S	units of $J(S)$	0^{th} element, $\_(S,0)$
m	ℝ	real number	all real numbers	integer	0.0
k	💡	concept	all information and concepts	concept	–
p	⌘	maps	all functions, functionals, etc.	map	–
n	ℤ	integer*	all integers	integer	0
s	☺	organism	all objects belonging to living entities	organism	I
t	⚡	charge	all spacetime infinitesimals of charged particles	coulomb	–
–	–	electric field	electric field magnitudes (∈ℝ)	<s,m,m,m>	–
b	𝔹	Boolean†	{FALSE, TRUE}	integer	FALSE
h	◎	soul/mind	all souls/minds	soul/mind	my mind
g	♡	feeling	all thoughts and feelings	feeling	–
ŋ	☊	body	all particles	body	my body
d	⌖	position	all spacetime infinitesimals of all objects	<m,m,m>	here
ʔ	∅	null	none	–	–
f	☼	internal energy	all spacetime infinitesimals of massive particles	joule	–
ʃ	🕓	time	all spacetime infinitesimals of all objects	second	now
r	∗	entropy	all spacetime infinitesimals of massive particles	joule	–
z	⊙	mass	all spacetime infinitesimals of massive particles	kilogram	–
–	–	gravitational field	gravitational field magnitudes (∈ℝ)	<s,m,m,m >	–
v	𝕍	vector‡	all vectors	integer	–
θ	⬡	thing	all particles	thing	–

* Integers are formed by "m" followed by any number of vowels. Each vowel, in the order shown in Table 1, corresponds to the digits 0-9, respectively.

† The two Booleans are “bi” for FALSE or “ba” for TRUE.

‡ Vectors are formed by applying the “v” function to any number of scalars.

field theory. Consequently, Complexlang would also allow two individuals (human to human, or perhaps even human to machine) with sufficient mathematical and scientific education to converse with very little additional learning. Similarly, Complexlang could be used for compact transfer and storage of information.

2. Phonetics

Phonemes for Complexlang (see Table 1 and Table 2) were greedily selected in order of the most prevalent phonemes among languages worldwide (Moran & McCloy). However, some were discarded due to similarity with previously selected phonemes.

Most of the selected vowels, i, y, ɯ, u, e, o, ɛ, ɔ, a, and ɑ (indicated hereafter using IPA), coincide with the IPA vowel gridlines, which benefit from high sound contrast. The other selected vowels, ɪ, ʊ, and ə, are also quite phonetically and spatially distinct. Additionally, functions which have a tendency to neighbor other vowels were assigned vowels which correspond to semivowels.

The each of the voiceless consonants, k, p, s, t, h, f, ʃ, and θ, were selected before its voiced counterpart, g, b, z, d, ɦ, v, ʒ, and ð, due to ease of articulation. No approximants were selected because these are easily confused with vowels.

3. Orthography

The orthography of Complexlang is phonemic (each written letter corresponds to a single phoneme and vice-versa). As shown in Table 1 and Table 2, Complexlang is simply written using the corresponding IPA symbols for their phonemes. Alternatively, the vowels can be written using their mathematical symbols and the consonants using emojis.

4. Morphology and Syntax

Like in most languages and mathematical expressions, Complexlang contains objects and relationships, can represent the order of those functions/relationships in a tree, and produces declarative statements which the speaker claims is true. In Complexlang, each consonant represents a set of elementary objects, while each vowel represents a Boolean, set, or quantitative operation. By applying operations to consonants, it's possible to create a variety of new sets. Each expression in Complexlang is necessarily a statement that the speaker purports as true. Upon forming an expression by applying operations to various objects, the speaker can express various beliefs.

Like any other language, since the structure is tree-shaped, we must serialize the tree to render it into a dictatable sequence. Like English and many other languages, we serialize using inorder traversal of the tree. Since inorder traversal is typically only defined for binary trees, to generalize to arbitrary trees, we define that half of the children (rounded down) of each branch lie in the left subtree, and the rest in the right. Figure 1 shows an example Complexlang expression tree which is then transcribed through inorder traversal into a sequence.

Because information is lost during inorder serialization, we cannot deterministically deduce the original tree from the spoken sequence. Hopefully, the tree structure can be deduced through context. However, if it cannot, then the structure can be clarified using pitches; the speaker can sing the expression by selecting a pitch for each vowel that is lower than the pitch for the vowels in higher branches.

Alternatively, preorder or postorder traversal would preserve enough information during serialization such that the tree structure can be recovered. However preorder traversal has a tendency to create consonant clusters which are difficult to articulate; in particular, consonant pairs are completely impossible to enunciate clearly. Vowel clusters are also common and more difficult to articulate. Postorder traversal suffers from the same problem. Additionally, the vowels (functions) appear

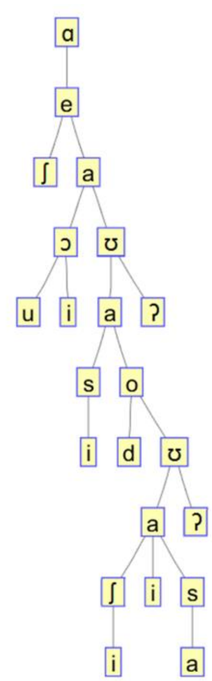

Figure 1. Example expression “ɑʃeuɔiasiadoʃiaisaʊʔʊʔ,” which translates literally to “or(deal(time;and(isGreaterThan(indexOfThis;zero);doesNotEqual(intersection(me; yourCurrentLocation);null))))” or roughly to “I will eventually be at your current location.”

at the end of the sequence, so the listener does not know which consonants (objects) belong to which functions until the end of the sentence, which can be very confusing. Also, preorder serialization is almost never and postorder serialization is rarely found among natural languages, which could also impose a barrier to learning.

We chose not to include any grammatical inflection to keep the linguistic rules simple.

Note that a formal distinction between morphology and syntax does not exist in Complexlang because each letter is already a complete sememe. Instead, speakers are encouraged to form words merely for convenience. For example, in Figure 1, one of the subtrees is "oʃiaisaʊʔ," literally translating to "subsetWhere(positions;doesNotEqual(intersection(now;the;you);null))," or roughly "your current location" (see Semantics). Thus it would be reasonable to define this as "your current location."

In writing, words are delimited by spaces and by pauses in speech.

5. Semantics

5.1. Sets (Consonants)

Each consonant in Complexlang corresponds to the set of all elements in the set of all objects of a certain subtype (see Table 2). However, the elements and nature of these objects may vary depending on the beliefs of the speaker. We do not claim to know all the answers to metaphysics. For example, "z," defined as mass, might represent the set of all elementary particles in the universe, or perhaps the set of all strings. A speaker may choose not to use "h," defined as souls/minds, if the speaker does not believe those exist. "s," defined as organisms, might represent the set of all particles belonging to living entities (indexed by individual entities). "ʃ," defined as time, might represent the set of everything (index by timed). Performing functions on these elementary sets lets us build more complex sets. For example, taking the intersection of the 0^{th} index of organisms (I) and the 0^{th} index of time (now) gives us just set of objects inside the person I am now.

The most frequent (or second most frequent in case of repeats, and so on) consonant in the word(s) corresponding to each set subtype among languages worldwide was chosen to represent the set subtype.

5.2 Set and Quantity (Units) Subtypes

Objects in Complexlang are typed. The supertypes are Boolean, set, and number, but these can have subtypes which inherit all properties of the supertype. Sets (consonants) and their elements have subtypes concept, organism, charge, etc., which are useful for selecting subsets (see Set Indexing). Set operations with inputs of different subtypes cast the output subtype as the subtype of the first input. Numbers can have the subtype integer, rational, real, complex, integer vector, rational vector, and so on.

5.3. Set Indexing

In order to build sets with appropriate subsets, sets are indexed depending on the set subtype (see the column in Table 2). If the same object belongs to multiple sets, it possesses multiple indices. For example, even though time and space both contain everything, the objects in time are indexed by moments (in units of seconds), while the objects in space are indexed by position (a 3D real vector in units of meters^3). To enable this scheme mathematically, all set-building functions, : and ¤, use the specialized definition, $Z(S) = \{\{x|x \in S, j(x) = i\}\big|i \in J(S)\}$, where $J(S) \subseteq \mathbb{K}^{\mathrm{n}}$ is the index set of S using the recommended units, $j: S \to J(S)$ is the index of the input element in S, and $k: S \times J(S) \to 2^S$ is the element(s) of S with input index. This has the effect of breaking the set into the disjoint union of subsets with overlapping indices of that particular set's subtype.

Because many of these sets are infinite, we sometimes needed to define default values in order to meaningfully select elements from sets and

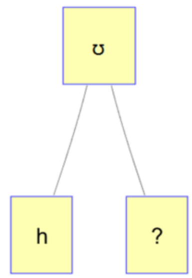

Figure 3. Example expression "hʊʔ," which translates literally to "souls exist."

express useful statements. Thus for many sets, the 0^{th} element is specifically defined (see the column in Table 2). Rather than using the # function, individual elements in a set can be indexed directly inserting an numerical sequence into the subtree. When inserted into the subtree of a set (consonant), the vowels i through ɪ, in the order shown in Table 1, correspond to the digits 0-9, respectively, in decimal. For example, "s(a(u(i)))" is the organism at index 120 (currently undefined).

Each set subtype which interacts with the # (measure) operator possesses an appropriate measuring function. For example, subsets of position (type set, subtype position) are indexed by meters^3 and are measured by taking the 3-dimensional Lebesgue measure of the index set of the subset, $\mu(J(x \in d))$, thus giving the 3-volume in units of meters^3 (type quantity, subtype real number, subsubtype meters^3). Subsets of charge are indexed by Joules of infinitesimal subunits of all charged particles in the universe. Thus the 1-dimensional Lebesgue measure of a charge subset, $\mu(J(x \in t))$, gives the amount of charge in Coulombs.

5.4 Scalars and Vectors

To form numbers explicitly (rather than simply using them to index sets), we must index the corresponding scalar set (e.g. integer, Boolean, etc.) using integer indices. Thus "n(a(u(i)))" refers to the integer at index 120 which corresponds, somewhat redundantly, to 120. If clarification is needed for larger numbers, each vowel can be paired with the corresponding consonant (in the order of Table 2) to improve audibility. Thus "n(a(k(u(p(i(m))))))" also corresponds to 120.

Like integers, Booleans are formed by the "b" function followed by vowels; "b(i)" is FALSE and "b(a)" is TRUE.

Rationals, reals, and other numbers can only be formed by performing functions on integers.

Vectors are formed by applying the "v" function to any number of scalars. For instance, "v(m(a)m(u)m(i))" represents [1;2;0].

5.5. Functions (Vowels)

Each vowel encodes a logical, set, and/or numerical operation (see Table 1). These behave as expected. For example, $\wedge(P,Q)$ (which would be transcribed as "p(a(q))" in Complexlang) is a Boolean expression which is true if both P and Q are true. $\#(S,I)$ is the subset $T \subseteq S: \forall s \in S(j(s) = I)$.

Since many logical operations have set and scalar algebraic analogs, we took advantage of polymorphism and overloaded these operators. The

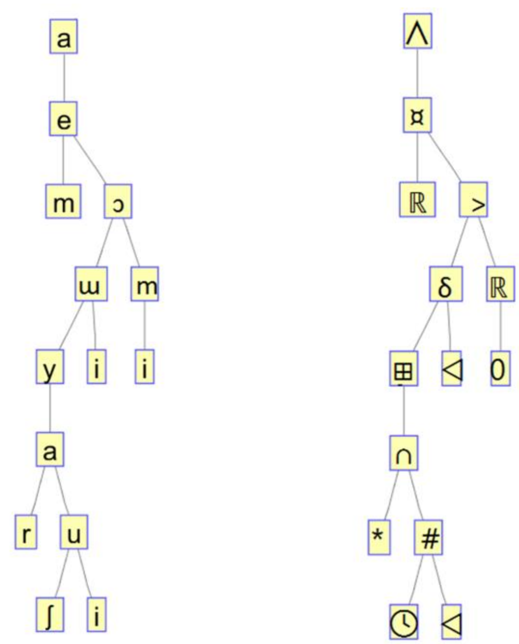

Figure 2. Example expression in Complexlang (left) and using emojis (right). It transcribes into "ameyraʃuiɯiɔmi," which translates literally to "and(deal(reals;isGreaterThan(differentialOf(measureOf(intersection(entropy;atIndex(time;it)));it); reals(zero))))" or "entropy increases with time."

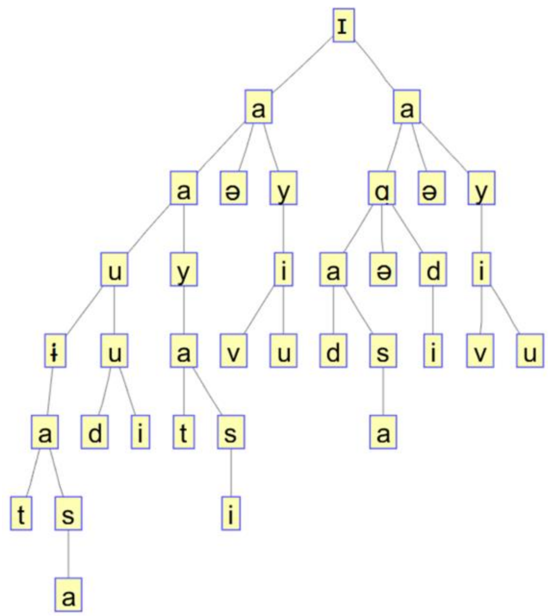

Figure 4. Example expression in Complexlang (top), also shown using emojis (bottom). The expression transcribes into "ɨtasaudui aytasiaəyviuıdasaɑədiaəyviu," which translates literally to "equals(times(times(atIndex(forceFieldOf(intersection(charge; you));atIndex(positions;zero));measureOf(intersection(charge; me)));inverse;sizeOf(the(vectors;two)));times(directionToYou; inverse;sizeOf(the(vectors;two))))." Mathematically, this can be written as $\boldsymbol{E_{tot}} \coloneqq \sum_{i \in t \cap you} \boldsymbol{E_{d(i)di}}, \boldsymbol{u} \coloneqq \boldsymbol{E_{tot}}(\boldsymbol{t} \cap \boldsymbol{me}) \cdot \boldsymbol{\mu}(\boldsymbol{J}(\boldsymbol{t} \cap \boldsymbol{me})), \boldsymbol{v} \coloneqq (\boldsymbol{d} \cap \boldsymbol{you}) - (\boldsymbol{d} \cap \boldsymbol{me}), \frac{\boldsymbol{u}}{|\boldsymbol{u}|} = \frac{\boldsymbol{v}}{|\boldsymbol{v}|}$. The rough translation is "you electrically attract me."

operator applies the corresponding function depending on whether its children are Booleans, sets, scalars, or operators. The output subtype is not always the same as the input.

6. Examples

In Complexlang, philosophical concepts (e.g. Figure 3, Figure 2, Figure 4) are easy to express. One minor disadvantage is that mundane everyday concepts of no interest, such as in Figure 5, are difficult to express.

Acknowledgements

The author would like to thank Cory Stevenson for numerous discussions, Junxing Wang for insight and discussions on data structures among other things, and Zack Coker for helping with the presentation.

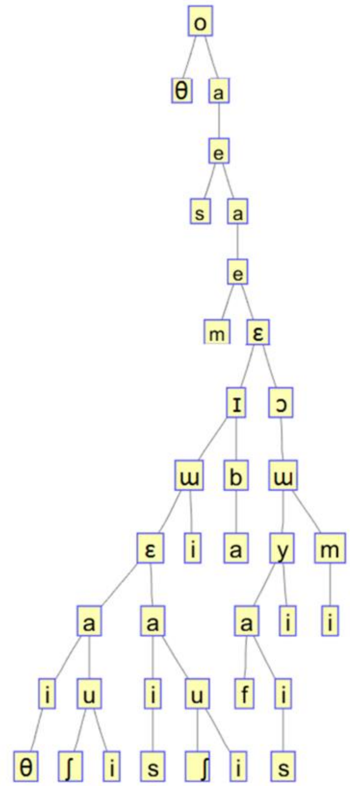

Figure 5. Example expression "θoaseameiθaʃuiɛisaʃuiɯiıbaɛɔfaisyi ɯmi," which roughly translates to "food." The literal translation is left as an exercise to the reader.

References

Bliss, C. K. (1965). *Semantography, a non-alphabetical symbol writing, readable in all languages; a practical tool for general international communication, especially in science, industry, commerce, traffic, etc., and for semantical education, based on the principles of ideog.* Sydney: Institute for Semantography.

Bourland, D. D., & Johnston, P. D. (1991). *To be or not: An E-prime anthology.* Institute of GS.

Cowan, J. W. (1997). *The complete Lojban language* (Vol. 15). Logical Language Group.

Karasev, I. (2006, March). *Arahau*. Retrieved 4 5, 2019, from Ivan Karasev's site:

https://sites.google.com/site/rbardalzo/conlangs/arahau-1

Lang, S. (2014). *Toki Pona: The language of good.* Tawhid.

Moran, S., & McCloy, D. (n.d.). *Segments*. Retrieved 2019, from Phoible 2.0: phoible.org/parameters

Quijada, J. (2004). Retrieved 2019, from Ithkuil: A Philosophical Design for a Hypothetical Language: http://www.ithkuil.net/

Weilgart, W. J. (1979). *aUI: the language of space, Pentecostal logos of love & peace: for the first time represented and adapted to the needs of this planet.* Cosmic Communication Co.

Zamenhof, L. L. (1887). *Dr. Esperanto's International Language: Introduction & Complete Grammar.*