Preface

道可道，非常道。
名可名，非常名。
無名天地之始;
有名萬物之母

The Path that can be trodden,
is not the eternal Path.
The Name that can be named,
is not the eternal Name.
Nameless, it is the Originator
of heaven and earth
Named, it is the Mother
of ten thousand things.
...
-Dao De Jing, chapter 1

Ana Bosnić

Contents

Chapter 1

Introduction

1.1   The story of King Atlas

1.2   Linguistic relativity

Max Geraerdts

1.3   Need for an IAL

Jonathan Roose

1.4   Eco's words

Jonathan Roose

Chapter 2

Phonology

2.1   The Sound of Atlan

Niek Elsinga and Stijn Janssens

Chapter 3

Writing Atlan

3.1 Typography and direction of reading

ATLAN'S writing system is a natural application of our philosophy: start with elementary parts, and every complexity will be a mere combination of those parts. Our glyphs (as we shall call them) each denote one syllable. They always do so; they always will stand for the *same* syllable. Unlike English: in the words "tone"and "to", the "to"is pronounced respectively [tʊɪ] and [tɵ].

That is the rationale behind our writing system; let us dive into the details. As told, Atlan has a set of basic lines. They are:

Consonants		Vowels	
Line	**In I.P.A.**	**Line**	**In I.P.A.**
|	...	∪	...
─	...	⊃	...
╯	...	∩	...
╲	...	C	...
╭	...	○	...
╯	...		
·	...		
○	...		

These lines all represent a single vowel (*V*), or a single consonant (*C*). We can combine them to make syllables. By combining two consonant lines, you get a CVC-syllable, such as *loj*, *pas* or *mup*. You can also make a VC-syllable, such as *mu*, *po*, or *ji*. The vowels don't have separate lines in a CVC or VC-syllable; instead, the vowel is determined by the position of the two consonant-lines. We will go deeper into that below. First, we give the rules for the order of the consonants and vowels: what determines whether two lines make e.g. *poj* or *jop*, *mu* or *um*?

This order is determined by the manner in which the lines combine. There is always a "bigger "line, and a

smaller one. These lines fit inside an imaginary box. The position of the smaller line relative to the bigger line, determines the order of consonants. A general rule of thumb is best given with the help of a box:
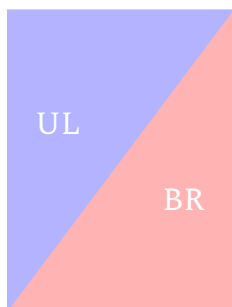
Figure 1: Box for determining consonant order.

If the smaller line is in the upper-left triangle (UL), it the consonant it designates comes first. If it is in the bottom-right one, it comes second. For the rest of the explanation, it is advised to keep this box in the back of your head. An example:
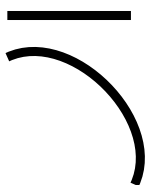
As you see here, the smaller line is found on top. Hence, it is placed inside the upper-left triangle. The consonant for which the smaller horizontal line stands (the *k*), comes before the other consonant, the *s*.
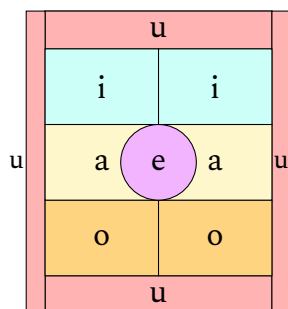
The vowel is…

- *u* if the smaller line is found at the edges. The smaller line is in its whole above, under, left or right of the main line.

- *i* if the smaller line is found on the upper-left or upper-right side of the main line. It is usually smaller than the line made for *u*, to avoid confusion.

- *a* if the smaller line is found left or right to the middle of the line.

- *o* if the smaller line is found on the bottom-left or bottom-right hand of the main line. Again, this line is smaller than the line for *u*.

- *e* if the smaller line is placed in the middle. Or, if the small line intersects with the main line at the middle. In some instances, the small line is then split up by the main line.

Then we have a single exception. You can combine two equivalent lines, to make syllables such as *pop*, *mum*, or *lol*. The order of these lines doesn't matter; hence we choose place the smaller line to the upper-left of the main line in such cases. For the vowel *u*, there are two small lines, split at the center. For *e*, there are either two or three small lines. At least one of those lines crosses through the center of the imaginary box.

Remember that the *p* is represented by the dot   ⋅   . For clarity, we couldn't combine simply two dots to make a full syllable. Hence,

These are the rules.

3.2   Numerals and Mathmatics

3.3   Font in T_EX
Jarno Smets

3.4   On Dyslexia
Stijn Janssens and Jonathan Roose

3.5   Cartouche

Chapter 4

Morphosyntax

4.1   Unambiguous Syntax
Jarno Smets

A MBIGUITY is of all times and places, and natural language is rife with it. *Goal*, *purple people eater*, *John trades with Mary*; these words and expressions can all be interpreted in multiple ways. Some despise ambiguity, while others wallow in it. Whatever one thinks of ambiguity, it is a part of natural languages.

For our constructed language, we want to minimize ambiguity. This for the sake of clarity and communicability. Hence this essay.

In this essay, I will cover a specific type of ambiguity, namely *syntactic ambiguity*, also known as *structural ambiguity*. A sentence that can be interpreted in multiple ways due to its syntax, is structurally ambiguous.

My aim in this essay is twofold. First, I want to show why syntactic ambiguity is a problem, especially for the goals of our project. Then, I will propose a strategy to minimize this form of ambiguity, and argue for that strategy choice.

What is syntactic ambiguity?

Syntactic ambiguity occurs when word-order gives rise to multiple interpretations (Oaks,2012, p.16) . The sentence "I see the man with binoculars" could be parsed (split into grammatical parts) in two ways:
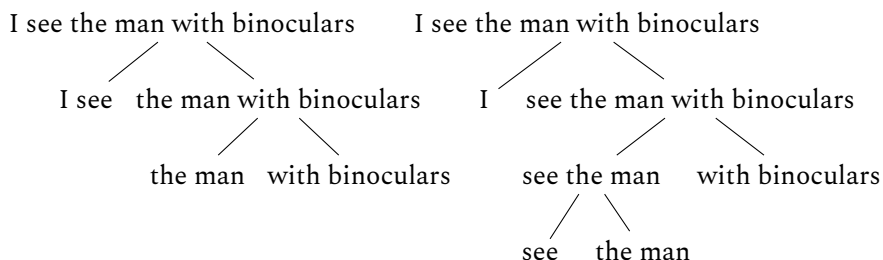
I see the man with binoculars ⟋⟍ I see the man with binoculars

I see   the man with binoculars   I   see the man with binoculars

the man   with binoculars   see the man   with binoculars

see   the man

Figure 1: different syntax trees for "I see the man with binoculars"
As we see in the above syntax trees, the difference in interpretation hinges on the (de)coupling of the words *man* and *with binoculars*. You could make *with binoculars* modify *man*. One could also modify *see* via *with binoculars*. The structure of the sentence doesn't give preference to one over the other.

For further illustration: one common type of syntactic ambiguity, is *scope ambiguity*. Scope ambiguity co-occurs mostly with logical operators such as quantifiers (for all, there exists), negation, and coordinators (and, or, but)[1]. *Scope* is the part of a sentence over which such

---

[1]These are all operators in propositional logic. Quantifiers: $\forall$ = for all , $\exists$ there exists, $\neg$ = negation, "not ". $\wedge$ = and/but, $\vee$ = or.

9

quantifier, negation, or coordinator ranges. Other instances of scope ambiguity are seen with modifiers, which I will briefly discuss below. Scope and scope ambiguity can best be explained by example:

(1) My cat is not grey or black

Two readings for (1): my cat is neither grey nor black, he is red, for example. Alternatively, my cat is not grey, but is black of colour. The scope for negation is ambiguous here. The *not* either or it has scope over *grey or black*, or it only has scope over *grey*, .

Where lies the origin of such structurally ambiguous sentences? Yang (2014) discerns five major causes of structural ambiguity in English:

A Negation scope

B Words with special syntactic functions

C Improper abbreviation

D Unclear word-characteristics

E Unclear modifier-relations

We discussed an instance of A above already. With B, Yang refers to words that generate *subordinate clauses*; subsentences. These sentences could either be the object of the bigger sentence, or be a truly subordinate clause.

E.g.: *The girls reported to me when they came.* Did the girls report to me after they arrived? Or did they report their time of arrival? it is unclear, due to the meaning of the word *when* .

Now on to cause C Yang mentioned. *Improper abbreviation* is the improper shortening of a sentence. Again, think of the sentence *Mary trades cards with Joe*. I could have said *Mary trades cards together with Joe* if I wanted to convey that message. But I didn't; I left out the word *together*, making it ambiguous.

Then, an example will elucidate cause D: *drinking water is unsafe*. Is *drinking* a verb in itself, or part of the larger phrase *drinking water*? The word characteristics for *drinking* are unclear. *Drinking* can either be seen as a verb, or as a noun together with *water*.

Finally, cause E refers to a modifier. A *modifier* is a linguistic element that changes the meaning of another linguistic element. For example, *grey* modifies *dog*. With unclear modifier relations, it is not apparent which modifier modifies what. In the phrase *purple people eater*, it is unclear whether *purple* modifies *people*, or *eater*.

I propose we bring these causes down to two. Firstly, structural ambiguity is caused by unclear semantic roles. A *semantic role* of a word or sentence-part is the role it plays in the meaning of the sentence. For example, the semantic role of *the grey dog* is the same in both sentences underneath:

(2) The cat attacked the grey dog
The grey dog was attacked

In the example given above, *I see the man with binoculars*, the semantic role of *with binoculars* is indeterminate. Is *with binoculars* how I see the man? Or does the man have binoculars? it is precisely this indeterminacy that seems to generate the ambiguity.

The second cause I propose, is unclear word-grouping and unclear scope. To get rid of the ambiguity in phrases as *purple people eater*, or *lesbian vampire killer*, it needs to be specified which words modify which.

The issue for Atlan

In the previous part, I examined syntactic ambiguity. Now, why is this a problem for Atlan?. I will here propose three reasons for that goal. First, I will argue that structural ambiguity inhibits the parsing of language by computers. Computer-parsing could boost the spread of Atlan. Secondly, I will show that some forms of syntactic ambiguity would endanger the communicative function of our constructed language. Atlan should be a bridge between two languages. Syntactic ambiguity can make it more difficult for two speakers from different languages to communicate. Lastly, I will argue that, in some high-stakes circumstances, syntactic ambiguity could be a great danger.

First of all, syntactic ambiguity is a problem for computers. Computers need a so-called *parser* to understand our language: The machines pick apart a sentence, in order to fully understand it (Schubert,2020). Syntactic ambiguity is a true roadblock for such parsing. Because syntactic ambiguity gives rise to multiple parsing options, a computer can't give a definite parsing of a syntactically ambiguous sentence. To circumvent, or to (partially) overcome it, multiple algorithms have been created. Yet it remains a difficult problem (Chowdhary, 2020, p.645).

For our constructed language, computer parsing and processing could be of help to the language learner. Translations would be more accurate, and practice materials can be generated more quickly. The presence of syntactic ambiguity is troubling for computers to analyze natural language.

Besides, structural ambiguity endangers universality. Since our language is intended as an auxiliary constructed language, people learn our constructed language as a *second language*. Hence, learners all approach our language from the perspective of their mother tongues. Now here lies the problem: different languages have interpret scope in different ways.

This has been shown, for example, in Scontras et al. (2017). This team of researchers found out that Mandarin lacks *inverse scope*. Inverse scope can best be explained by an example: "A badger dug every hole ". In English, two readings are available for such sentence:

Surface scope - *There was one badger such that it dug every hole.*
Inverse scope - *For every hole, there was a (different) badger that dug it.*

Scontras et al. found out that the inverse scope reading is simply not available in Mandarin Chinese. Furthermore, they found out this lack of inverse scope is found in the English of native Mandarin speakers. Another study showed similar results: Korean learners of English habitually preferred the surface-scope reading, and left the inverse-scope reading out (Seon & Shin, 2022).

So, when learning new languages, speakers have the tendency to bring their native scope-reading preferences with them. This endangers the communicative function of our conlang. If our constructed language has certain scope ambiguities in it, miscommunication can occur. Say you have speaker X, in whose language both scope-readings are available. She communicates such a scope-ambiguous sentence to speaker Y. X wants to bring across the inverse scope-reading. To speaker Y, *inverse* scope-readings are *not* available. Then X fails to bring across *her* wished interpretation of the sentence; a communicative error has occurred. Hence, structural ambiguity endangers the communicative clarity of our constructed language.

Expanding further on communicative clarity: some contexts strictly demand that there be no ambiguity. Hazardous environments, such as nuclear power plants, weapon

factories and the like, should communicate in a clear, unambiguous manner. Also law practice should be ridden of ambiguity. These are high-stake-environments. Any communication mistake could have far-stretching consequences.

Say an English nuclear-power plant has the following instructions etched into an important control panel:

(3) In case of emergency: pull the horizontal striped lever

Now, there are two levers in the control room. One is a lever you pull from north to south, and it is marked with horizontal stripes. The other lever is horizontal, but has vertical stripes instead. Which lever do you pull? I hope this example makes it clear how dangerous syntactic ambiguity can be.

Of course, this was a fabricated example. A real-life example, can be found in (Layman,1962):

(4) Serbian subjects in the United States, shall enjoy the rights which the ... laws grant ... to the subjects of the most favoured nation.

Example (4) elicits two interpretations: Serbian subjects who reside already in the United States enjoy the rights, or Serbian subjects, independent of where they remain, enjoy the rights when they are in the United States. This is syntactic ambiguity in law. Here it can have grave consequences for a large number of citizens.

With these few examples, I have shown why syntactic ambiguity is best left out in our constructed language. Firstly, it would make it hard for computers to parse our language. That while computers generally help to spread a language faster. Secondly, structural ambiguity in a language can cause miscommunication within a language. Not every language allows multiple scope readings, for example. Scope ambiguity can then lead to miscommunication in a language. Thirdly, syntactic ambiguity can be of real danger. It could cause communicative issues in high-stakes environments, such as infrastructure and law.

It must be noted, however, that syntactic ambiguity is not only a *bad* phenomenon. It can also serve poetic and humorist endeavours. For example, the structurally ambiguous sentence

(5) Time flies like an arrow; fruit flies like a banana

is undeniably witty[2]. Does the fruit fly similar to a banana, or do fruit flies love a banana? The first part of (5) seems to prime the reader for the first reading.

Minimizing syntactic ambiguity

Now I will look at the efforts of other constructed language to minimize syntactic ambiguities. I will examine the benefits and downfalls of their approaches. From

---

[2]Found in (Cornish-Bowden, 2015).

that examination, I will aim to distill the strategy for *our* constructed language to bring structural ambiguity to a minimum.

One of the main origins of structural ambiguity is the distance between sentence-parts. In a structurally ambiguous sentence, it becomes unclear how the words are fit into phrases, and then how phrases fit in a sentence. For example, in the noun-phrase *purple people eater*, does *purple* belong to *people*, or to *eater*? Solving structural ambiguity is then making clear which words modify what , to only give one interpretation of a phrase or sentence.

The constructed language Lojban[3] indeed does this. It has two ways of specifying which words belong together. The first manner comes in the form of the structure word *bo*. *Bo* enforces scope (The Lojban Reference Grammar, 2023). To see how, let's take the English sentence "That is a big bug catcher". In English, you could interpret this either as a big catcher of bugs, or a catcher of big bugs. In Lojban, the word *bo* makes this difference explicit:

(6) That is a bug-catcher that is big.
Ta barda miptera bo kavbu[4].

---

[3]Lojban [loʒban] is a constructed language, created by a group of people wanting to improve another constructed language, *Loglan*. One of its spear points is having an ambiguous syntactic structure. Found on: https://mw.lojban.org/papri/Lojban, may 23rd, 2023.

[4]*Ta* = "That is", *barda* = "big", *miptera* = "bug", *kavbu* = "catcher", and *bo* is the structure word. English translation found in (Jbovlaste: a lojban dictionary, 2023)

(7) That is a catcher of big bugs.
Ta barda bo miptera kavbu.

As you might have guessed from the above examples, the structure word *bo* "pulls"two words together, to combine them. Since the combination of words is made explicit by *bo* , ambiguity is resolved.

There is a second way of coupling words in Lojban. The makers of Lojban decided to make rules for grouping, the so-called *brivla*. *Brivla* is an umbrella term for nouns, verbs, adjectives and adverbs (The Lojban Reference Grammar, 2023). The *left-grouping-rule* states that the two leftmost *brivla* are grouped together. So, the sentence *Ta barda miptera kavbu*, is automatically parsed equivalent to the second reading above (The Lojban Reference Grammar, 2023).

It seems Lojban got structural ambiguity under control with these two restrains. What are the advantages and disadvantages of this approach?

As already mentioned above, the word-groupings are made explicit, effectively removing structural ambiguity from the language. This increases the clarity of Lojban, and thereby makes the language more universal. There are some downsides however. As we saw above, some scope readings are not even available in the mother-tongue of some speakers. The left-grouping rule described above could enforce a reading upon the language learner, which the language learner is far from familiar with. Lojban then might sometimes give rise to miscommunications.

Another constructed language with the intent of minimizing (syntactic) ambiguity, is *Ithkuil*. Ithkuil marks semantic roles explicitly in noun cases (Ithkuil, Case Morphology, 2023). This is relatively similar to German, where the case *der* usually marks the (male) subject of the sentence, or *des* marks the possessor. Ithkuil has more cases, including the ones we all know (subject, object, possessor, dative). Examples are *instrument*, *force*, *agent*, and much more[5].

Ithkuil specifies the exact case of every noun. Due to that, it is clear which word plays what role in a sentence. In *purple people eater*, for example, *eater* could be nominative, while *purple people* would be marked as accusative. In that way, ambiguity is brought down to a minimum. However, there is one big downside to this approach: it is too complex. Ithkuil is very complex, and hard to learn. Even the creator, John Quijada, can't speak it fluently (Foer, 2023). Thus, the ubiquitous presence of cases seems to do more harm than good; it eliminates ambiguity, but at the cost of learning-ease and fluency.

We have seen how Ithkuil and Lojban deal with syntactic ambiguity. Taking this in account, how will Atlan deal with it?

A feature of Lojban was the explicit word-coupling with the structure word *bo* . The word directly made clear what words formed a separate noun-phrase. How-

---

[5]Readers interested in more should visit Ithkuil's website: http://www.ithkuil.net/newithkuil_04_case.htm.

ever, it is an extra word to remember. I believe it is a better idea to couple words in the most direct sense of the word: literally connect them to each other. This is a familiar feature of, for example, Dutch: *grijze honden-temmer* (grey hound-tamer) versus *grijze-honden temmer*. Both in English and Dutch, the words "dog "and "tamer "are joined to indicate that they belong together. In speech, words that should be separated, are separated by a pause.

Now, what about scope ambiguity? For negation, for example, we will include two types: sentential and predicate negation. Sentential negation is a form of negation that spans over a whole sentence. E.g. *I have **not** been to school today*. Predicate negation on the other hand, only spans over a predicate. For example, *I'm very **un**happy at the moment.* This would fix negation scope ambiguity. Take the aforementioned example *my cat is not grey or black*. The two readings can be separated using the distinction between types of negation:

(8) My cat is ungrey or black
it is not the case that my cat is grey or black

The sentential negation will take the form of a distinct particle, whereas the predicate negation will be an affix. This has the following reasons. Sentential negation spans over a whole sentence. To make it immediately apparent that a sentence is negated, it would be convenient to have a loose particle to place at the beginning of a sentence. Predicate-negation occurs within a sen-

tence, and binds to predicates. Hence, it will be an pre-
fix, connected to the predicate it negates.

This approach to negation doesn't make it more dif-
ficult to learn. Most languages are familiar with it: th
most common types of negation are negative particles,
and affixes (Martin et al., 2005, p. 454) Even if, for a
learner's mother-tongue, there is a mismatch between
negation type (sentential and negation) and form (par-
ticle and affix), the forms are very likely familiar. This
will very likely make our approach to negation somewhat
more intuitive for a language learner. Moreover, predi-
cate negation is present in a majority of languages (Mar-
tin et al., 2005, p.467).

But what about scope ambiguity outside of negation?
E.g. *The dog or the cat and the bird made a mess.* Here, we
appeal to operator strength from Classical Logic. Nega-
tion comes first. Then comes conjunction ("and"). Last
comes disjunction ("or") (O'Donnell et al., 2007, p.120)[6].
In the above example, the sentence is read as: (the dog or
the cat) and (the bird) made a mess. That the bird made a
mess, is certain. Whether the dog or the cat made a mess
is uncertain.

Now it is worth noting a few *caveats* about my ap-
proach. Firstly, I reasoned mostly from syntactic ambi-
guities in English and Dutch. This could leave room in

---

[6]After that comes the conditional ("if...then", →) and the bi-
conditional ("if and only if ", ↔). As far as I can tell, they don't seem
to generate syntactic ambiguity, hence I leave them unmentioned
here.

my solutions for syntactic ambiguities not thought of by me. Hence, I talked primarily of *minimizing* syntactic ambiguity. Besides, it is worth noting that context will disambiguate as well. I have mostly examined structurally ambiguous phrases and sentences in isolation. Some of those phrases or sentences would not be as ambiguous in context.

In this essay, I have shown two things. First, I argued that syntactic ambiguity should be avoided when constructing a language. This because syntactic ambiguity troubles computers, endangers communicative function, and can be potentially harmful.

Secondly, I have proposed several general recommendations for battling syntactic ambiguity. This I distilled from previous attempts at constructing structurally unambiguous languages, such as Lojbans and Ithkuil . Lojban made its structure clear, but had a redundant syntax rule. Ithkuil explicitly specified the semantic role of each word, but became extremely hard to learn and speak as a consequence.

Atlan won't be as specific as Ithkuil or Lojban. It is a balance we need to find between preciseness and learnability. Both Ithkuil and Lojban are extremely precise, but sacrifice learnability. I am confident that Atlan will find a good balance, and that the learner will profit from that.

Chapter 5

Our Ontology

Chapter 6

Lexicon

6.1 Language Families
Jep Antonisse

6.2 AI Generation
Max Geraerdts and Jep Antonisse

6.3 Atlan to English

6.4 Translation Protocol

6.5 Synonyms

Chapter 7

Dictionary

Chapter 8

Pragmatics

8.1 Logophilia, Wordplay and Rhythm

8.2 Mood-markers

Chapter 9

Further suggestions

9.1 Music Notation

9.2 Sign Language

9.3 Language Variation
Niek Elsinga

9.4 Lingo

Chapter 10

Example texts

10.1   The Story of Babel
Jonathan Roose

10.2   Alice in Wonderland
Jarno Smets

10.3   Declaration of Universal Human Rights
Max Geraerdts

10.4   The Epic of Gilgamesh
Jep Antonisse

10.5   Punishment of Atlas
Stijn Janssens

Epilogue

Bibliography

Index