Lojban++:
An Efficient, Minimally Ambiguous, User-Friendly Natural-Like Language for Human-Computer, Computer-Computer and Human-Human Communication

Ben Goertzel
August 22, 2006

Thanks are due to Debbie Duong, Moshe Looks and Nick Stone for edits and suggestions; and to Arnt Richard Johansen and Robin Lee Powell for copious help with the Lojban examples.

1 Introduction

The goal of this article is to briefly and loosely describe a new language called Lojban++. Which, as the name suggests, is not really all that new, being a modification (and in some senses, it is proposed, an improvement) of an existing language, Lojban.

The primary goal of Lojban++ is to constitute a language for efficient, minimally ambiguous, and user-friendly communications between humans and suitably-constructed AI software agents. (Because the term "AI software agent" is somewhat long and cumbersome, for the rest of this article we will use the briefer term "bot" instead.) Along the way it will be noted that Lojban++ may also be utilized for communications between (suitably trained) humans, or for communication between different bots.

Some details on the particulars of the Lojban++ language proposal, aimed at readers familiar with Lojban, are given in the Appendix. The body of the article describes Lojban++ and proposed related R&D work at a more abstract level.

2 Lojban vs. Lojban++

Lojban is itself an outgrowth of another constructed language, Loglan, created by Dr. James Cooke Brown around 1955 and first widely announced in a 1960 *Scientific American* article (Brown, 1960). Loglan is still under development but now is not used nearly as widely as Lojban. First separated from Loglan in 1987, Lojban is a constructed language that lives at the border between natural

language and computing language. It is a "natural-like language" in that it is speakable and writeable by humans and may be used by humans to discuss the same range of topics as natural languages. Lojban has a precise, specified formal syntax that can be parsed in the same manner as a programming language, and it has a semantics, based on predicate logic, in which ambiguity is carefully controlled. Lojban semantics is not completely unambiguous, but it is far less ambiguous than that of any natural language, and the careful speaker can reduce ambiguity of communication almost to zero with far less effort than in any natural language. On the other hand, Lojban also permits the speaker to utilize greater ambiguity when this is desirable in order to allow compactness of communication.

Consider the English sentence,

```
When are you going to the mountain?
```

When written in Lojban, it looks like:

```
do cu'e klama le cmana
```

In Lojban++, with the judicious importation of English vocabulary, it takes a form more recognizable to an English speaker:

```
you cu'e go le cmana
```

A fairly standard predicate logic rendition of this, derived by simple, deterministic rules from the Lojban++ version, would be

```
atTime( go(you, mountain), ?X)
```

Many individuals attempting to learn and use Lojban have found, however, that it has two limitations. The Lojban vocabulary is unfamiliar and difficult to learn – though no more so than that of any other language belonging to a language family unfamiliar to the language learner. And, more seriously, the body of existing Lojban vocabulary is limited compared to that of natural languages, making Lojban communication sometimes slow and difficult.

To address these issues, I have previously proposed a Lojban variant called Loglish (Goertzel, 2005a, 2005b), which consists of a combination of Lojban syntax and selected Lojban vocabulary, extended with English vocabulary. Lojban++, as proposed here, may be understood as a pidgin of Lojban and Loglish, advocating more extensive use of Lojban vocabulary than did the original Loglish proposal. Lojban++ is less elegant than Lojban but significantly easier to learn, and much easier to use in domains to which Lojban vocabulary has not yet been extended. In short, the goal of Lojban++ is to combine the mathematical precision and pragmatic ontology that characterize Lojban, with the usability of a natural language like English with its extensive vocabulary.

Consider the more complex English sentence,

```
When are you going to the small obsidian mountain?
```

In Lojban, there is no word for obsidian, so one needs to be invented (perhaps by compounding the Lojban words for "glass" and "rock," for example), or else a specific linguistic mechanism for quoting non-Lojban words needs to be invoked.

But in Lojban++ one could simply say,

```
you cu'e go le small obsidian mountain
```

The construct "small obsidian mountain" is what is called a Lojban *tanru*, meaning a compound of words without a precisely defined semantics (though there are recognized constraints on tanru semantics based on the semantics of the components (Nicholas,1996)).

Alternatively, using the Lojban word, marji, which incorporates explicit place structure (*x1= material/stuff/matter of composition x2*), a much less ambiguous translation is achieved:

```
you cu'e go le small mountain poi marji loi obsidian
```

in which "poi marji loi obsidian" means "that is composed of [a mass of] obsidian."

This illustrates the flexible ambiguity achievable in Lojban. One can use the language in a way that minimizes ambiguity, or one can selectively introduce ambiguity in the manner of natural languages, when desirable.

3 Communicative Practice, Linguistic Theory and Software Development

The detailed development of Lojban++, beyond the elements described in the Appendix, will involve an ongoing interplay between three critical aspects: communicative practice, linguistic theory, and software development.

3.1 Communicative Practice

Communicative practice is both fundamental and time-consuming. Both Lojban's ontology and its linearization aspects have been refined extensively through its use in practice among the Lojban community. This kind of

The differences between Lojban and Lojban++ are subtler than it might appear at first. It is key to understand that Lojban++ is not simply a version of Lojban with English character-sequences substituted for Lojban character-sequences.

A critical difference lies in the rigid, pre-determined argument structures associated with Lojban words. For instance, the Lojban phrase

```
klama fi la .atlantas. fe la bastn. fu le karce
```

corresponds to the English phrase

```
that which goes from Atlanta to Boston by car
```

To say this in Lojban++ without using "klama" would require

```
go fi'o source Atlanta fi'o destination Boston fi'o vehicle car
```

which is much more awkward. On the other hand, importing more Lojban into the Lojban++ expression, one obtains

```
klama fi la Atlanta fe la Boston fu le car
```

which may more optimally balance simplicity with familiarity. The point is that the Lojban word "klama" comes with the convention that its second argument (indexed by "fi") refers to the source of the going, its third argument (indexed by "fe") refers to the destination of the going, and its fifth argument (indexed by "fu") refers to the method of conveyance. No such standard argument-structure template exists in English for "go", and hence using "go" in place of "klama" requires the use of the "fi'o" construct to indicate the slot into which each of the arguments of "go" is supposed to fall.

This example illustrates the kind of tuning that may be done to optimize Lojban++ based on practice and experience, leveraging the decades of experience that have already gone into Lojban.

refinement-via-practice buys a level of maturity that cannot be obtained in a shorter period of time via formal analysis or creative invention. For example, the current Lojban treatment of quantifiers is arguably vastly superior to that of any natural language (Cowan, 1997), but that was not true in 1987 when it excelled more in mathematical precision than in practical usability. The current approach evolved through a series of principled revisions suggested from experience with practical conversation in Lojban. Any new natural-like language that was created for human-bot or bot-bot communication would need to go through a similar

process of iterative refinement through practical use to achieve a similar level of refinement.

To be maximally effective, the development of Lojban++ must continue the pragmatic communicative tradition that has characterized Lojban. The history of Lojban shows that this is an invaluable approach to refining linguistic mechanisms and finding out which theoretically attractive ideas *really* work.

3.2 Linguistic Theory

Next, linguistic theory is the foundation of Lojban and is a valuable source for ideas to be tested and refined via communicative practice. An extensive formal treatment of Lojban grammar has been published (Cowan, 1997), and while there is no published hard-copy Lojban dictionary, there is a website (jbovlaste.lojban.org/) that serves this purpose and which is frequently updated as new coinages are created and approved by the Logical Language Group, a standing body charged with the maintenance of the language.

Regarding the theoretical development of Lojban++, three areas that we see meriting particular focus are quantifiers, the semantics of compositions, and reference resolution. Lojban's system of quantifiers has evolved significantly, but key aspects such as the interrelationship of universal and existential quantification have not been that thoroughly tested in practice and may admit refinement. Regarding the semantics of composition, one of the challenges for AI bots interpreting Lojban++ statements will be assigning appropriate meanings to tanru (unstructured word combinations such as "small obsidian mountain"). Use of tanru can be avoided but this sometimes leads to excessively long utterances. Heuristics may be developed for tanru interpretation, but it is critical to theoretically explore the semantic principles underlying such heuristics.

Finally, the Lojban system for referencing syntactically distant entities (the analogue of English "it", "he" and so forth) is widely recognized as imperfect, and the current Lojban++ proposal contains some modifications to it that require exploration both on the theoretic and pragmatic levels

3.3 Developing Parsing and Semantic Mapping Software

In order that Lojban++ be useful for human-bot communications, parsing and semantic mapping software need to be produced for the language.

There is a fully functional Lojban parser based on a parsing expression grammar (Powell, no date specified), as well as an earlier parser based on BNF grammar. (And, parenthetically, the observation that Lojban is more conveniently formulated in PEG grammar form is in itself a nontrivial theoretical insight.) The creation of a Lojban++ parser based on the existing Lojban parser, is a necessary and a relatively straightforward though not trivial task.

On the other hand, no software has yet been written for formal semantic interpretation ("semantic mapping") of Lojban expressions – which is mainly

because Lojban has primarily been developed as an experimental language for communication between humans rather than as a language for human-bot communication. Such semantic mapping software is necessary to complete the loop between humans and AI reasoning programs, enabling powerful cognitive and pragmatic interplay between humans and bots. For Lojban++ to be useful for human-bot interaction, this software must be created and must go in both directions: from Lojban++ to predicate logic and back again. As Lojban++ is a superset of Lojban, creating such software for Lojban++ will automatically include the creation of such software for Lojban proper.

Compared to the construction of a Lojban++ syntax parser, the construction of tools for semantic interpretation of Lojban++ promises to be a little subtler, requiring integration of existing linguistics resources such as WordNet (Feldbaum, 1998) and Longman's English Dicionary (an excellent commercial dictionary available in electronic form). In this vein, it is important to understand that – as noted in the inset box above -- the difference between Lojban and Lojban++ is not just one of renaming Lojban words with more familiar names. Rather, Lojban words come with specific argument-structures which allow more compact expressions, while Lojban++ sacrifices some compactness for increased familiarity – a fact that makes semantic mapping of Lojban++ significantly subtler.

4 Lojban++ and Artificial Reasoning

One of the most attractive features of Lojban and Lojban++ is how straightfowardly they can be translated into predicate logic format such as is currently used by a variety of AI reasoning systems. This means that as soon as Lojban++ semantic mapping software is constructed, it will almost immediately be possible to communicate in a meaningful way with bots embodying logical reasoning engines.

This capability of Lojban has already been explored in a preliminary way by Speer and Havasi's (2004) JIMPE software application, which involves a semantic network guiding logical reasoning, Lojban parsing and Lojban language production. The inset box below shows an excerpt from a sample conversing-and-reasoning session with JIMPE. While JIMPE is a relatively simplistic prototype application, it is clear that more complex example of Lojban-based artificial inference are also relatively straightforwardly achievable via a conceptually similar methodology.

In order to better understand this aspect of Lojban++, we are currently exploring some preliminary experiments involving translating Lojban++ statements into predicate logic, and then having the Novamente AI Engine carry out reasoning

coi mi'e jimpe	*Hello, I'm* JIMPE.
> ganai la bab crino	If Bob is green,
gi la erik blanu	then Eric is blue.
je'e	*Okay.*
> la erik na blanu	Eric is not blue.
je'e	*Okay.*
i ua la bab na crino	*Aha! Bob is not green.*

Sample Session with JIMPE, a prototype Lojban-based reasoning system
from (Speer and Havasi, 2004)

on these logic statements, and then translating Novamente's conclusions back into Lojban++. In one such experiment, we feed Novamente the Lojban++-derived predicates given in the inset box on the following page, corresponding to the English sentence

```
Hey, I just saw a bunch of troops going into the woods.
What do you want me to do?
```

We also give Novamente the knowledge that it is hard to see things in forests, and some other simple related background knowledge. The Novamente system then concludes that it is currently difficult to see where the soldiers are.

Our preliminary conclusion from Speer and Havasi's work plus our own experiments is that it is highly viable to feed knowledge into an AI reasoning system using Lojban++, have the system derive conclusions from this knowledge, and then translate the conclusions back into Lojban++ for human consumption. In short: Lojban++-based communication and collective cognition between humans and bots. The key point is that the Lojban or Lojban++ forms may – once appropriate semantic mapping software is written, as we propose -- be automatically translated into and out of predicate logic form, which may be freely manipulated by AI systems. On the other hand, to translate an English sentence into predicate logic form requires a complex combination of processes including word sense disambiguation, part of speech tagging, syntax parsing and semantic mapping. Mapping a simple English sentence into predicate logic is just barely within the capability of current natural language processing systems; and when presented with more complex sentences, current NLP systems fail, whereas the corresponding process for Lojban or Lojban++ remains simple and straightforward. For instance, the sentence you are currently reading, when fed to any existing natural language processing, would give rise to a variety of different parses (mostly conceptually undesirable), each leading to several different semantic mappings (mostly conceptually undesirable, even when based on a conceptually desirable syntactic parse) – whereas the Lojban or Lojban++ cognate of this sentence would give rise to a single syntactic parse with a small variety of semantic mappings.

Because of its easy convertibility into and out of predicate logic, Lojban++ should provide a effective mechanism for human-bot communication of commonsense knowledge and information about the physical, psychological and social worlds -- and for communication between bots, as well. In this vein it is worth noting that Lojban/Lojban++ contains two distinct aspects:

1) an ontology of predicates useful for representing commonsense knowledge (represented by the Lojban *cmavo* along with the most common Lojban content words)
2) a strategy for linearizing nested predicates constructed using these cmavo into human-pronounceable and -readable strings of letters or phonemes.

The second aspect is of no particular value for communication between bots, but the first aspect is. We suggest that it makes sense for bots dealing with commonsense and real-world knowledge to communicate using a "tree form" of Lojban++ that embodies the semantics of Lojban++ utterances but is expressed as a mathematical tree rather than a linear sequence of characters or phonemes. Conversion between tree form and character-string form is simple and automatic but is necessary only for human-bot communication, not for bot-bot communication.

Relatedly, we suggest that the Lojban++ ontology provides a useful framework for knowledge representation that may be incorporated at a fundamental level into any AI system that centrally utilizes predicate logic or a similar representation. Following up Speer and Havasi's prototype work, we intend to utilize this ontology within the Novamente AI Engine (Goertzel, 2006), an AI system that controls a humanoid bot acting in a 3D simulation world (Goertzel et al, 2006).

The English sentence

```
Hey, I just saw a bunch of troops going into the woods. What
do you want me to do?
```

(drawn from Bruno, 2006) translates into the Lojban

```
ju'i do'u mi pu zi viska lo nu so'i lo sonci cu nenkla le
ricfoi .i do djica lo nu mi mo
```

or the Lojban++

```
Hey do'u mi pu zi see lo nu so'i lo soldier cu enter le
forest .i do  want lo nu mi mo
```

which literally transcribed into English would be something like

Hey! [vocative terminator] I [past] [short time] see an event of (many soldiers enter forest). You want event (me what?)

Omitting the "hey," a simple and accurate predicate logic rendition of this sentence would be

```
past( $X)
short_time($X)
$X = see(me, $Y)
$Y = event( enter( $Z, forest) )
soldier($Z)
many($Z)

want(you, event( ?W(me) )
```

where *?W* refers to a variable being posed as a question be answered, and $X and so forth refer to internal variables.

The Lojban and Lojban++ versions have the same semantics as the predicate logic version, but are much simpler to speak, hear and understand due to the lack of explicit variables and Lojban's other convenient syntactic features.

English	I eat the salad with croutons
Lojban	mi citka le salta poi mixre lo sudnabybli
Lojban++	mi eat le salad poi mixre lo crouton mi eat le salad poi contain lo crouton

English	I eat the salad with a fork
Lojban	mi citka le salta sepi'o lo forca
Lojban++	mi eat le salad sepi'o lo fork

English	I will drive along the road with the big trees
Lojban	mi litru le dargu poi lamji lo barda tricu
Lojban++	mi ba travel fi'o vehicle lo car fi'o route le road poi adjacent lo so'i big tree mi ba litru fi lo car fe le road poi adjacent lo so'i big tree mi ba drive fi'o route le road poi adjacent lo so'i big tree

English	I will drive along the road with great care
Lojban	mi litru le dargu ta'i lo nu mi mutce kurji
Lojban++	mi ba drive fi'o route le road ta'i lo nu mi much careful mi ba litru le road ta'i lo nu mi much careful

English	I will drive along the road with my infrared sensors on
Lojban	mi ba litru le dargu lo karce gi'e pilno le miktrebo'a terzga
Lojban++	mi litru le road lo car gi'e use le infrared sensor mi litru le road lo car gi'e pilno le infrared te zgana mi drive fi'o vehicle lo car fi'o route le road gi'e use le infrared sensor

English	I will drive along the road with the other cars
Lojban	mi litru le dargu fi'o kansa lo drata karce
Lojban++	mi ba drive fi'o route le road fi'o kansa lo so'i drata car mi ba drive fi'o route le road fi'o with lo so'i drata car mi ba litru le road fi'o kansa lo so'i drata car

Further Lojban++ examples,
showing multiple variant uses of Lojban++ mechanisms where appropriate

5 A Lojban++ Software Suite

Above I mentioned the need for syntax parsing and semantic mapping software for Lojban++. In addition to these necessary tools, a number of other tools would also be useful and desirable to maximize the utility of Lojban++. The following list is fairly comprehensive, including both necessaries and nice-to-haves:

1. A **Lojban++ parser** based on modification of the current Lojban parser, with incorporation of WordNet, FrameNet and statistical NLP resources as appropriate
2. **Automated Lojban++ semantic mapping software**, which incorporates the current Lojban parser and carries out a complete mapping from Lojban text to predicate logic and vice versa
3. The creation of **software to measure the speed of human knowledge encoding and knowledge comprehension** in Lojban++
4. The creation of **software to measure the precision of knowledge encoding** achievable using Lojban++ by trained humans within various periods of time. For this we intend to use logical-inference-based precision measures: the premises of inferences will be entered into a logical reasoning system using the languages, and adequate precision will be judged by whether the reasoning system can use these premises to draw appropriate conclusions. A set of benchmark inference cases will be chosen based on relevance to the needs of bots in the national and homeland security domain.
5. The creation of **speech-to-text software for Lojban++**, utilizing open-source HMM speech-to-text software that has been tuned for English and making appropriate modifications (for instance the Sphinx or Julius open-source speech engines would be appropriate for this purpose)
6. The creation of initial prototype **software to translate between Lojban++ and English**, using existing NLP resources as appropriate. Note that this software is expected to be highly imperfect, and the perfection of such translation software is a larger project going beyond the scope of this proposal.

Several of the key software components involved in the above points, and their interconnections, are illustrated in Figure 1 below.

With the creation of these tools – or even the first two -- the line of research begun by Dr. James Cooke Brown in 1955 will have taken a major leap forward. Lojban++ will be a fully-developed, scientifically-validated natural-like language for human-human, human-bot and bot-bot communication, with all necessary software support. As artificial intelligence software advances during the coming years, such a linguistic framework will be an extremely major asset.

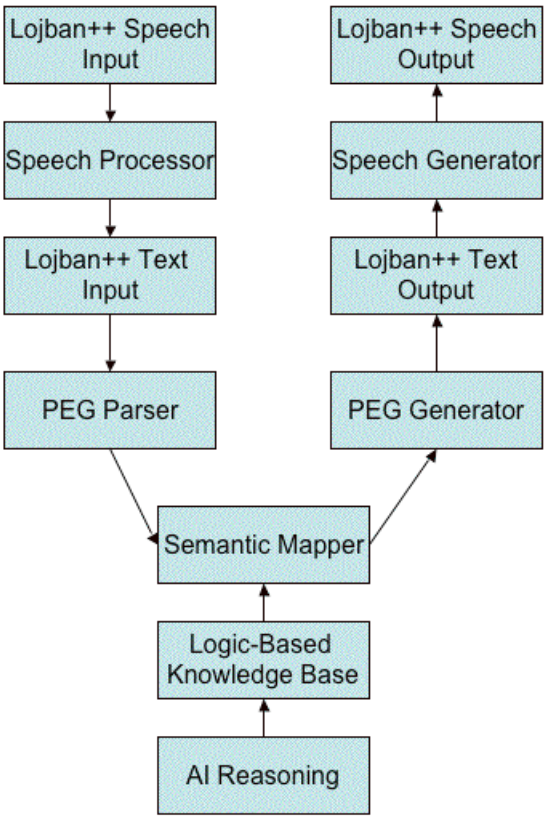

Figure 1

Diagram of Key Software Components in Proposed Lojban++ Processing Framework

References

- Brown, James Cooke (1960). Loglan. *Scientific American*, June 1960
- Bruno, Greg (2006). Talking with machines: Scientists see new language as key to robotic revolution. *Times Herald-Record*, July 20, 2006
- Cowan, John Woldemar (1997). *The Complete Lojban Language*, The Logical Language Group
- Feldbaum. Christine (1998). *Wordnet: An Online Lexical Resource*. MIT Press.
- Goertzel, Ben (2005). *Potential Computational Linguistics Resources for Lojban.* http://www.goertzel.org/new_research/lojban_AI.pdf
- Goertzel, Ben (2005a). *Loglish*, online at
- http://www.goertzel.org/new_research/Loglish.htm
- Goertzel, Ben (2005b). *LojLink: Design for a Loglish-Based Knowledge Management System*, online at
- http://www.goertzel.org/new_research/LojLink.pdf
- Goertzel, Ben (2006). Patterns, Hypergraphs and General Intelligence. *Proceedings of International Joint Conference on Neural Networks,* IJCNN 2006, Vancouver CA (available online at www.novamente.net)
- Goertzel, Ben, Hugo Pinto, Ari Heljakka, Michael Ross, Izabela Goertzel, Cassio Pennachin (2006). Using Dependency Parsing and Probabilistic Inference to Extract Gene/Protein Interactions Implicit in the Combination of Multiple Biomedical Research Abstracts, *HLT-NAACL Workshop on Bio-NLP, New York NY*
- Goertzel, Ben, Ari Heljakka, Moshe Looks, Cassio Pennachin. "Exploring Android Developmental Psychology in a Simulation World," in *Proceedings of the Workshop on Android Science, International Conference on the Cognitive Sciences*, Vancouver 2006
- Goertzel, Ben, Matthew Ikle', Izabela Freire Goertzel and Ari Heljakka (in preparation). *Probabilistic Term Logic*
- Nicholas, Nick (1996). Semantic issues in a prescriptive word composition theory, online at http://www.tlg.uci.edu/~opoudjis/Play/lojban.html
- Nicholas, Nick (2002). Folk Functionalism in Planned Language: The Long-Distance Reflexive vo'a in Lojban. *Journal of Universal Language* 3:1. 133-167.
- Nicholas, Nick (1996a). Lojban as a Machine Translation Interlanguage in the Pacific. *Fourth Pacific Rim International Conference on Artificial Intelligence: Workshop on 'Future Issues for Multilingual Text Processing'*, Cairns, Australia, 27 August 1996. 31-39.
- Nicholas, Nick and John Cowan (2003). *What Is Lojban?*, The Logical Language Group
- Nicholas, Nick and Robin Turner (date unspecified). *Lojban for Beginners*, online at http://ptolemy.tlg.uci.edu/~opoudjis/lojbanbrochure/lessons/book1.html

- Powell, Robin Lee (date unspecified). *Issues with the Lojban Formal Grammar*, online at http://www.digitalkingdom.org/~rlpowell/hobbies/lojban/grammar/index.html
- Speer, Rob and Catherine Havasi. *Meeting the Computer Halfway: Language Processing in the Artificial Language Lojban*, Proceedings of MIT Student Oxygen Conference, 2004, http://sow.lcs.mit.edu/2004/proceedings.html
- Wang, Pei (2006). *Rigid Flexibility: The Logic of Intelligence*, Springer-Verlag

Appendix 1:
Some Lojban++ Specifics

This Appendix reviews the key principles by which Lojban++ incorporates English words into Lojban, and discusses some other small additions that Lojban++ makes to Lojban. It is intended mainly for readers who are familiar with Lojban.

A note is perhaps appropriate here regarding the right approach to learning Lojban++ at present. Lojban++ is a variant of Lojban, and no systematic teaching materials for Lojban++ yet exist. Therefore, at the moment the only way to learn Lojban++ is to learn the basics of Lojban, and then learn the differences (note however, that the "basics of Lojban" as meant here does not necessarily include a broad mastery of Lojban vocabulary beyond the cmavo or "structure words"). Assuming Lojban++ is developed as proposed here, relevant teaching materials should also be developed, such as Pimsleur-style language CD's (or audio files) for Lojban++ and a "Conversational Lojban++" text. There is no need to write a book-length grammar for Lojban++ comparable to (Cowan, 1996), however, since the prinicples of Lojban++ grammar are all drawn from Lojban.

Finally, a necessary caveat: Lojban++ is not yet refined through practice, so it should be assumed that the specifics described in this Appendix are likely to be subjected to change through experience, as the language is used and developed.

Basic Principles for Using English Words in Lojban++

This list of principles will likely be extended and refined through usage.

1. Content words only! English words that are about syntactic relationship have no place in Lojban++.
2. No "being verbs" or "helping verbs." The English "is" and its conjugations have no place in Lojban++, for example.
3. All Lojban++ cares about is the main part of an English word. None of the English markers for tense, person or number should be used, when importing an English word into Lojban++. For instance, English verbs used must be in the infinitival form; English nouns must be used in the singular form. For instance, "run" not "runs" or "ran"; "pig" not "pigs."
4. English adverbs are not used except in rare cases where there is no adjectival form; where there is an adjectival form it is used instead – e.g. "scary" not "scarily."

To lapse into Lojban lingo, English words must be used in Lojban++ as brivla. Tense, number and so forth are supposed to be added onto these brivla using the appropriate Lojban cmavo. The Lojban++ parser will assume that any

non-Lojban word encountered, if not specifically flagged as a proper name (by the cmavo “la”), is an English word intended to be interpreted as a brivla. It will not do any parsing of the word to try to interpret tense, number, adverbiality, etc.

Next, English idiomatic collocations, if used in written Lojban++, should be used with an underscore between the component words. For example: New_York, run_wild, big_shot, etc. Without the underscore, the Lojban++ parser will assume that it is seeing a tanru (so that e.g. “big shot” is a type of “shot” that is modified by “big”). In spoken Lojban, the formally correct thing is to use the new cmavo “quay” to be discussed below; but in practice when using Lojban++ for human-human communication this may often be omitted.

Finally, a less formal guideline concerns the use of highly ambiguous English words, the use of obscure senses of English words, and the use of English words in metaphorical senses. All of these should be avoided. They won’t confuse the Lojban++ parsing process, but they will confuse the Lojban++ semantic mapping process. If a usage seems like it would confuse an AI program without much human cultural experience, then try to avoid it. Don’t say

```
you paint ti
```

when you could say

```
you cu vivid bo describe ti
```

The latter will tell an AI exactly what’s happening; the former may leave the AI wondering whether what’s being depicted is an instance of description, or an instance of painting with an actual paintbrush and oils. Similarly, to say

```
you kill me
```

when you mean

```
you much amuse me
```

is not in the Lojban++ spirit. Yes, an AI may be able to figure this out by reference to dictionaries combined with contextual knowledge and inference, but the point of Lojban++ is to make communication simple and transparent so as to reduce the possibility for communication error.

Syntax-based Argument Structure Conventions for English Words

Next, one of the subtler points of Lojban++ involves the automatic assignment of Lojban argument-structures to English words. This is done via the

following rules:

- Nouns are interpreted to have one argument, which is interpreted as a member of the category denoted by the noun
    - *la Ben human*

- Adjectives/adverbs are taken to have two arguments: the first is the entity modified by the adjective/adverb, the second is the extent to which the modification holds
    - *la Ben fat le slight*

- Intransitive verbs are interpreted to have at least one argument, which is interpreted as the argument of the predicate represented by the verb
    - *le cockroach die*

- Transitive verbs are interpreted to have at least two arguments, the subject and then the object
    - *la Ben kill le cockroach*

- Ditransitive verbs are interpreted to have three arguments, and conventions must be made for each of these cases, e.g.

    - give x y z may be interpreted as “x give y to z”
        - *la Ben give le death le cockroach*
    - take x y z may be interpreted as “x takes y from z”
        - *la Ben take le life le cockroach*

    A rule of thumb here is that the agent comes first, the recipient comes last, and the object comes inbetween.

Semantics-based Argument Structure Conventions for English Words

The above syntax-based argument-structure conventions are valuable, but not sufficiently thorough to allow for fluent Lojban++ usage. For this reason a collection of semantics-based argument-structure conventions have been created, based mostly on porting argument-structures from related Lojban words to English vocabulary. The following list is the current working version, and is likely to be extended a bit during actual usage.

- Plant or animal (moss, cow, pig)
    - x1 is a W of species x2
- Spatial relation (beneath, above, right, left)
    - x1 is in relation W to x2, in reference frame x3

- Dimension-dependent spatial descriptor (narrow, deep, wide, etc.)
  - x1 is W in dimension x2, relative to standard x3
- Unit (foot, hour, meter, mile)
  - x1 is x2 W's by standard x3
- Kinship or other interpersonal relationship (mother, father, uncle, boss)
  - x1 is the W of x2
- Thought-action (remember, think, intuit, know)
  - x1 W's x2
  - x1 W's x2 about x3
- Creative product (poem, painting, book)
  - x1 is a W about plot/theme/subject/pattern x2 by author x3 for intended audience x4
- Physical action undertaken by one agent on another (touch, kick, kiss)
  - x1 (agent) W's x2 with x3 [a locus on x1 or an instrument] at x4 [a locus on x2]
- W denotes a type of substance, e.g. mush, paste, slime
  - x1 is a W composed of x2
- Instance of communication (ask, tell, command)
  - x1 W's x2 with information content x3
- Type of utterance (comment, question)
  - x1 (text) is a W about subject x2 expressed by x3 to audience x4
- Type of movement (walking, leaping, jumping, climbing)
  - x1 (agent/object) W's to x2 from x3 in direction x4
- Route, path, road, trail, etc.
  - x1 is a W to x2 from x3 via/defined by points including x4 (set)
- Nationality, culture etc.
  - x1 reflects W in aspect x2
- Type of event involving humans or other social agents (celebration, meeting, funeral)
  - x1 partakes, with purpose x2, in event x3 of type W
- Posture or mode of physical activity of an embodied agent (stand, sit, lie, stoop)
  - x1 W's on surface x2
- Type of mental construct (idea, thought, dream, conjecture, etc.)
  - x1 is a W about x2 by mind x3
- Type of event done by someone, potentially to someone else (accident, disaster, injury)
  - x1 is a W done by x2 to x3
- Comparative amount (half, third, double, triple)
  - x1 is W of x2 in quality x3
- Relation between an agent and a statement (assert, doubt, refute, etc.)
  - x1 W's x2
- Spatial relationship (far, near, close)
  - x1 is W from x2 in dimension x3

- Human emotion (happy, sad, etc.)
  - x1 is W about x2
- A physically distinct part of some physical object, including a body part
  - x1 is a W on x2
- Type of physical transformation (e.g. mash, pulverize, etc.)
  - x1 [force] W's x2 into mass x3
- Way of transmitting an object (push, throw, toss, fling)
  - x1 W's object x2 to/at/in direction x3
- Relative size indicator (big, small, huge)
  - x1 is W relative to x2 by standard x3

Lojban gismu of clear use within Lojban++

There are some Lojban gismu (content words) which are clearly much more useful within Lojban++ than their English counterparts. Mostly this is because their argument structures involve more than two arguments, but occasionally it is because they involve a two-argument structure that happens not to be well-captured by any English word (but is usually represented in English by a more complex construct involving one or more prepositions).

A list of ~300 gismu currently judged to be "essential" in this sense is at http://www.goertzel.org/papers/gismu_essential.txt , and a list of <50 additional gismu judged potentially very useful but not quite so essential is at http://www.goertzel.org/papers/gismu_useful.txt .

Special Lojban++ cmavo

Next, there are some special cmavo (structure words) that are useful in Lojban++ but not present in ordinary Lojban. A few more Lojban++ cmavo may be added as a result of practical experience communicating using Lojban++; but these are it, for now.

qui

("kwee")

qui is a cmavo used in Loglish to create words with unambiguous senses, as in the example:

pig qui animal

pig qui cop

The second English word in the compound is a sense-specifier. Generally this

should only be used where the word-sense intended is not the one that would be most obviously expected given the context.

In some rare cases one might want two modifiers, using the form

(English word) qui (English word) qui (English word)

it / quu

("kwuhh")

The basic idea is that there is one special referential word in Lojban++ – "it" – which goes along with a reference-target-indicator "quu" which gives a qualitative indication of the referent of a given instance of "it," intended to narrow down the scope of the reference resolution process.

For instance, you could say

la Dr. Benjamin Goertzel cu proceed le playground. It quu man cu kill le dog. It cu eat le cat.

In this case, "it" is defined to refer to "Dr. Benjamin Goertzel", not to "man" generically. The "man" qualifier following the "quu" is intended to merely guide the listener's mind toward the right antecedent for the pronoun. It's not intended to explicitly define the pronoun. So, basically

it quu male

is the rough equivalent of the English "he", and

it quu female

is the rough equivalent of the English "she"

him/her/they

For sake of usability, it is worthwhile within Lojban++ to introduce the following shorthands

him ==> it quu male
her ==> it quu female
ver ==> it quu person
they ==> it quu people

(Note that "him" in Lojban++ thus plays the role of both "him" and "he" in English.)

quay

(“kway”)

This cmavo separates parts of an English collocation in speech, e.g.

```
big quay shot
```

It may often be omitted in informal speech; and in writing may be replaced by an underscore (big_shot).