Norman Friis

Lojban Orthography

History and Origins of Lojban: Lojban is a constructed language related to Loglan created by the Logical Language Group (LLG) in 1987. James Cooke Brown, the creator of Loglan, had copyrighted Loglan so the LLG sought to circumvent this copyright by using a different vocabulary on top of the underlying grammar and structure of Loglan. After a lengthy court battle the copyright was invalidated, but regardless, Lojban's supporters had already adopted this new vocabulary and named their language *Lojban: A realization of Loglan*.

Features of Lojban: Much like Loglan, Lojban is based on predicate logic. There are no irregularities in grammar or spelling. The language was designed to be easy to learn and culturally neutral. For these reasons, it lends itself to applications in artificial intelligence, machine translation of natural language text, and other similar linguistic and computer science fields.

Lojban is a synthetic SVO and sometimes SOV language. While Lojban's grammar is based upon and very similar to Loglan, there are minor differences. In addition, Lojban's grammar is formally defined using the tool YACC (with some formal "pre-processing" rules) which can create the code for a parser given a grammar written in BNF (Backus-Naur form) notation.

Orthography: Currently Lojban uses the standard Latin alphabet without *h, q, w* but with the following three letters: **,***(syllable separator)* **.***(pause, or glottal stop)* **'***(θ)*

In an effort to create an orthography which reflects some of the intended uses of Lojban, each character encodes within it some meaningful information as to its articulation. This is achieved by a simple 4 row, 2 column matrix. Starting with the top left element and moving left to right in a row major order, the elements are numbered from 0 to 7. (See the other page of the handout for a visual.) Each element can be 'on' or 'off' and this is represented (in script form) with a dot. Connect the dots as long as they are adjacent except for dot 7, it is never connected to adjacent dots.

The meanings for each element (when they are 'dotted') are as follows:

0 – stressed (only with vowels, and only if the word uses an unusual stressing pattern)
1 – non-sonorant
7 – end of word (see below)

<u>1 = 0: Non-sonorant:</u>
2 – bilabial/labial
3 – velar/post velar
2+3 – alveolar
4 – fricative
5 – stop
6 - voiced

1 = 1 and 6 = 0: Sonorant:
2 – bilabial/labial
3 – velar/post velar
2+3 – alveolar
4 – fricative
5 – stop
6 – vowel

1 = 1 and 6 = 1: Vowel:
2 – close
3 – open
2+3 – mid
4 – front
5 – back
4+5 – central

Each character has two (or four in the case of vowels which have stressed and non-stressed versions) possible realizations, one normal and one where the character appears at the end of a word. When element 7 is marked it takes the place of the space following the end of a word (this is different than the character '.' which means a pause or glottal stop). This is attractive because it cuts down on space used by text when represented on computers as strings of bits.

Consider the following line of Lojban text in traditional Latin alphabet:
.i le do nobli turni be la ter. ku se cfari

There are a total of 43 characters (including spaces), of those 10 are spaces (~23%). To represent this using ASCII it would take 344 bits (43 * 8 bits), of which 80 bits are effectively wasted space used only to help mark word boundaries.

By incorporating a way of marking word boundaries in the characters it is possible to reduce the number of bits required to represent this string (264 bits with the above proposed orthography). This may be an attractive feature considering potential applications with machine translation, storage and analysis of large bodies of text.