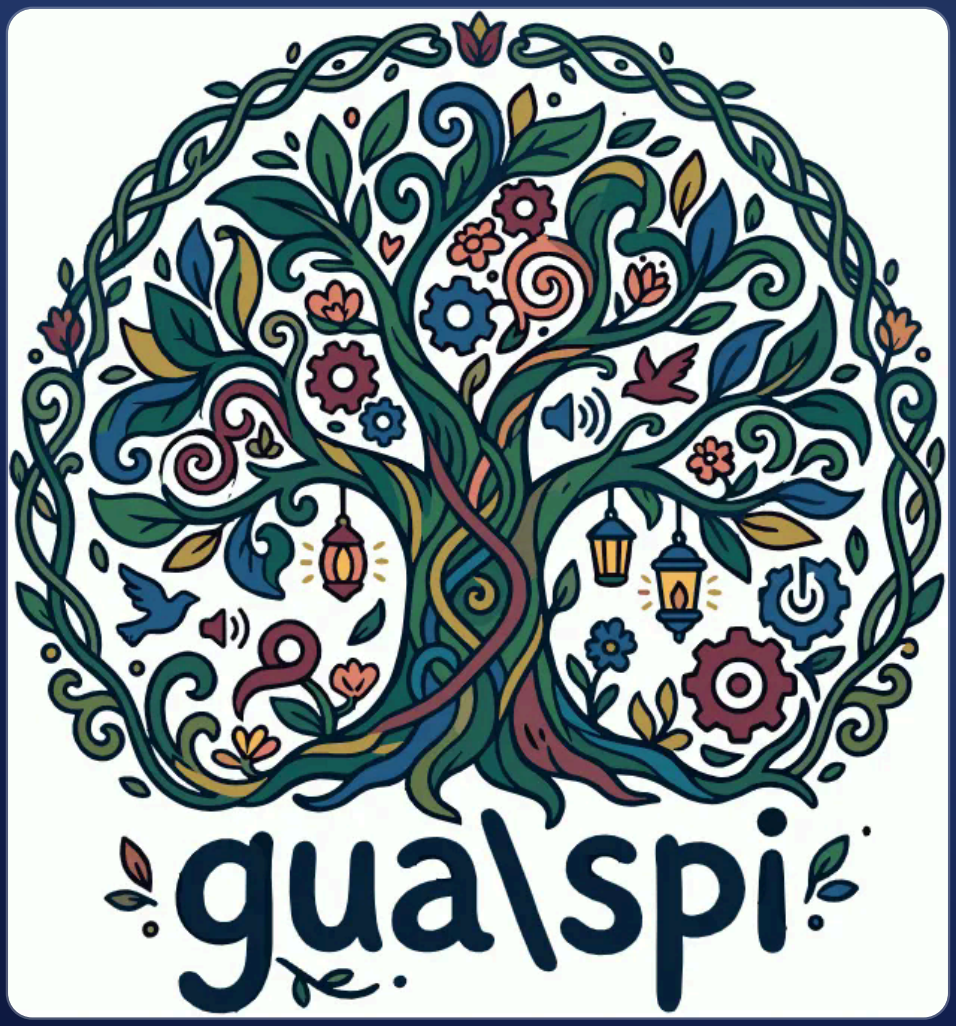

Gua\spi

James F. Carter — artificial language reference (mirrored web edition)

Gua\spi Index

James F. Carter

Abstract:

Gua\spi

is an artificial language suited to both humans and machines. It can express real human conversation. Yet the vocabulary and the grammar are two and three orders of magnitude simpler than English. Word and phrase meanings are defined through predicate calculus and hence can be represented and manipulated efficiently and unambiguously by programs (and people).

If you would like to modify the vocabulary CGI script for another language, use this link.

To download the dictionary in machine readable form, use this link.

Gua\spi, an Artifical Natural Language

James F. Carter

15 September 1991

Abstract:

Gua\spi

is an artificial language suited to both humans and machines. It can express real human conversation. Yet the vocabulary and the grammar are two and three orders of magnitude simpler than English. Word and phrase meanings are defined through predicate calculus and hence can be represented and manipulated efficiently and unambiguously by programs (and people).

Gua\spi is a language artifact. It can do everything that can be done by a natural language like English, though it is wholly artificial. But since it is so much simpler, studies of the phenomenon of language can also be simpler, more conclusive, and less costly in the investigator's time, and software which functions by imitating human language behavior can be written more easily.

The demonstration that *gua\spi* is fully functional as a natural language involves a difficult process: the reader must think of some language behavior, it is translated into *gua\spi*, and the reader then judges whether the translation is adequate, using his or her skill in *gua\spi* to understand the result. This paper hasn't enough length to be a complete users' guide to *gua\spi*, but I will present many of the basic concepts so the interested reader can make an independent judgement of whether simple phrases are adequately translated. Then I will give a short sample of live discourse, machine-translated from *gua\spi* to pseudo-English as an example of the kind of topics that can be represented and how much of the meaning survives mechanistic parsing. I hope the reader may be lead to wonder how various advanced topics are handled in *gua\spi* and that, on getting more complete information, he or she will find the treatment satisfactory.

Salient characteristics of *gua\spi* that particularly suit it to use by humans and artificial intelligences together are:

- *Gua\spi* is simple. The formal syntax is stated in a few lines in the Appendix, compared to thousands of lines for English, and the content words number only about 1400, compared to about half a million for English. Every valid utterance can be parsed in only one way.
- *Gua\spi* is efficient. Words are short, and extensive defaults on articles and modal cases eliminate the majority of structure words. Thus humans find it easy to speak.
- *Gua\spi* is modular. Morphology, grammar, organization, semantics and vocabulary are specified separately and interact to the minimum feasible extent.
- *Gua\spi* is complete. The content words form a basis such that almost any meaning not tied to a specific place or culture, and many which are, can be represented by agglutination. Foreign words and scientific Latin are welcome.

The language artifact *Loglan*, developed by James Cooke Brown [L1], was the inspiration for *gua\spi*. Brown realized that a very small set of content words could form a basis of a language, and produced such a set. By successfully writing large amounts of prose in *Loglan* while creating almost no additional words, I validated his insight. *Loglan* is almost 100 times simpler than English, and I have simplified the deep structures laid bare in *Loglan* by almost 100 times more to give *gua\spi*.

The description of *gua\spi* that follows is necessarily incomplete. I hope that the reader will be led to wonder how *gua\spi* handles this or that problem — and that, on getting more information, he will find that the problem is handled adequately. A fuller description of *gua\spi* is available, and a dictionary and teaching materials are in preparation [Ga].

Morphology — What is a Word

The phonemes (sounds) are divided in two classes, C's and V's, or *kona* and *vumu* in *gua\spi*. All C's are consonants in English and those English vowels used in *gua\spi* are all in the V class, hence the names. In addition each word has a tone (*dinu*), a frequency modulation of the V's of each word in the Chinese manner. A word is written as a tone (see Table 3 [Tones]), one or more C's and one or more V's. What could be simpler?

C/V	Length	Sound	Labial	Dental	Palatal	Velar	Glottal
C	Plosive	unvoiced	p	t	c*	k	—
C	Plosive	voiced	b	d	j	g	:*
C	Spirant	unvoiced	f	s	q*	—	—
C	Spirant	voiced	v	z	x*	—	#*
V	Vowels		u	o	y	i,e	a
V	Nasal etc.		m	n	l	w*	r

Table 1 [Phonemes]. *Gua\spi* phonemes, arranged by tongue position front to back (reading across) and sound type (reading down). Letters marked * differ from European standard usage.

Gua\spi	English	Examples of Pronunciation
c	ch	CHew, Ciao (Italian)
q	sh	SHoe
x	zh	aZure, breZHnev (Russian)
:	(pause)	the:apple, hawai:i (glottal stop)
#	uh	thE, Among (schwa)
u	u, oo	flUte, bOOt
o	o, oa	bOne, bOAt
y	i	knIt
i	i, ee	grEEn machIne (not eye)
e	e	bEd
a	a	fAther (not cAt)
mnlr	mnlr	LeMoN RiNd (no silent R)
w	ng	stroNG

Table 2 [Pronunciation]. How to pronounce *gua\spi* phonemes. Nonstandard C's are shown; C's without examples are as in English. Standard radio broadcast accent is close to correct for the V's; Spanish is closer. Pronounce the vowels as one sound, not a glide between two sounds as in *"eye"*.

Table 1 [Phonemes] shows the phonemes, categorized by tongue position and sound type. Some phonemes are represented confusingly in English, e.g. *'sh'* which sounds like neither *'s'* nor *'h'*. So in *gua\spi* they are assigned individual letters which differ from English usage — *'q'* for *'sh'*. Table 2 [Pronunciation] gives examples of these, and all the vowels. There is a 1-1 relation between written and spoken *gua\spi*. Written blanks have no sound, and are optional. There is no distinction between upper and lower case.

All words must begin with a C and end with a V so foreign words must be modified to fit. They appear in *gua\spi* as possibly several C^nV^n syllables. A prefix, most commonly **qo** for a foreign name, signals that the syllables are not to be looked up in the dictionary and are to be kept together when other compound words are split up during organization.

Grammar by Tones — How Words Join

The grammar is stated in Backus-Naur form in the Appendix. The job of grammar is to stick words together into phrases (*zdua*). *Gua\spi* grammar produces a strict tree-structure parse for each sentence wherein one phrase is at the root forming the main sentence, and each phrase has an ordered list of zero or more sub-phrases. The grammar does not support meaning of any kind — no tenses, no possessives, no nouns, no verbs. These ideas are handled at the organizational and semantic levels, using the grammar as a foundation. Like its morphology, the grammar of *gua\spi* is nearly minimal.

For grammatical purposes there is only one kind of phrase (though distinctions are made at the organizational level), but words have five categories: the two grammar words **fu** and **fi**, sentence start prefixes, other prefixes, and everything else. The main part of a phrase is a sequence of one or more words collectively called the *"phrase predicate"*; any prefixes in this must come first. During grammatical analysis there may be several content words in the predicate, though later transformations split up all compound words into separate phrases. After any of the prefixes or after the whole predicate the sub-phrases are interspersed where convenient; they attach to the current phrase at the next higher level.

Fig. 1 [Parsetree] shows a parse tree for a simple sentence with two levels of sub-phrases. The tones (see Table 3 [tones]) show the level of each word relative to the one before it.

> **^:i !tara /vme -crw !kseo ^vu -tum !kfor ^fe -fnau** *The rat devours (violently eats) the cheese with a fork and knife*

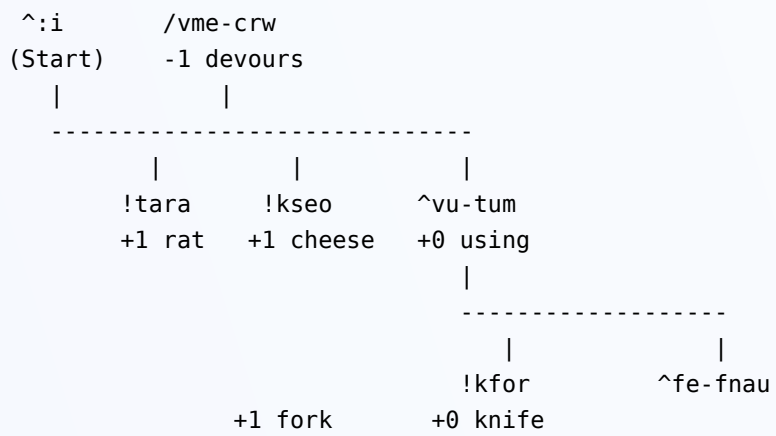

Figure 1 [Parsetree]. A Sample Parse Tree. Each phrase is part of the most recent one at the next higher level. Each phrase may have sub-phrases. Tone symbols (see Table 3 [Tones]) show jumps into and out of sub-phrases; they show the difference in level between adjacent words.

The tone, or frequency modulation, of a *gua\spi* word represents its parse tree level relative to the word just before it. Table 3 [Tones] shows the sounds and interpretations of the tones. The tones are the most terrifying aspect of *gua\spi* for speakers of European languages. Informal experimentation shows that naive non-Asian listeners can hear the tones reliably. Please remember that over a billion people in China and Southeast Asia speak tonal languages. If they can do it, so can you.

Number	Sound	Level	Type	Symbols	
1	High-even	Predicate	Compound	⚙	
2	Rising	One higher	Phrase	/	/
3	Down-up	One lower	Clause	|	⚙
4	Falling	One lower	Argument	\	!
5	Up-down	No change	Phrase	^	@
6	Low-even	Predicate	Transitive	=	%

Table 3 [Tones]. Sounds and interpretations of the tones. *"Level"* refers to the parse tree level of the word with that tone, relative to the structure before it. *"Type"* indicates the organizational type of that word or phrase. The first set of symbols shown, ASCII characters, is preferred but the second set can substitute on a manual typewriter. In this paper, **!** is used instead of * * for convenience in typesetting.

Tones - and = join adjacent words of a compound phrase predicate. Tones **!** and | start a sub-phrase of the current phrase. Tone ^ closes the current sub-phrase and starts a new one, part of the same containing phrase. Tone / closes a sub-phrase and resumes the predicate of the containing phrase, if among its prefixes, or otherwise starts a new phrase at the higher level. Distinctions within these tone classes are important later but do not affect the grammar.

A sentence start prefix such as **^:i** is automatically at root level, and **/fi** jumps to root level without ending the sentence. Other multi-level upjumps are available with **fu** but are needed rarely.

Cases — Members of a Relation

The next layer of *gua\spi* syntax is the organizational level, but to understand the reason for some organizations we have to make a detour into semantics to find out about cases (*skam*). We also get to see some examples of *gua\spi* sentences. To minimize vocabulary we will use variations on this one:

> **^:i !tara /crw !kseo** *The rat eats the cheese (sentence: rat eat cheese)*

Please pronounce it correctly: *'c'* as English *'ch'* and *'i'* as *'ee'*. Mind the tones, lest you change it into *"the eat rats the cheese"* or some such. (Chinese is worse: you could change *"mother"* into *"horse"* with a wrong tone. But Chinese people survive nicely.) The notation "tara-*rat*" in examples means that **tara** is the example word, and it means *"rat"* in English. Isolated words or phrases like this are written without a tone because it depends on the context where the word is used.

Natural languages generally distinguish between *"things"* and *"actions"*, where an *"action"* is a relation between *"things"*. The formal term for such a relation is a *"predicate"* (*gna*). Take for example:

> **^:i !tara /crw !kseo** *The rat eats the cheese*

"crw-*eats*", called a *"predicate word"* (*qury*), is a symbol for the predicate by which the rat and the cheese are related. The predicate is like a function whose arguments are things that might be related; the value of the function is true or false (or fuzzily in between) depending on whether or not they actually are thus related: in this sentence, whether the first actual parameter eats the second.

The formal parameters of a predicate, regarded as a function, are referred to as *"cases"*. English has *"nominative"* and *"accusative"* cases (the rat occupies the nominative case and the cheese occupies the accusative case), and Latin has in addition *"genitive"*, *"ablative"* and others, but *gua\spi* simply numbers the cases. Some *gua\spi* words have as many as five numbered cases. In our example, *"*tara-*rat"* fills the first case of *"*crw-*eats"* and *"*kseo-*cheese"* fills the second. Natural languages and -gua!spi have obvious regularities in how particular cases are used, but it is not possible, at least in -gua!spi, to make a universal theory of what cases mean. Users should attend closely to case patterns in related words, but each category must be learned individually.

The words denoting the actual parameters of a predicate are called *"arguments"*; being sub-phrases, they have their own predicate words. Here, **!tara** and **!kseo** are the arguments. The *"thing"* represented by an argument, which is the actual parameter of the sentence predicate, is something that can fill the first case of the argument's predicate. It is referred to as the *"referent"* of the argument. For example,

> **crw !kseo** *Eater of cheese*

is an argument phrase; the first case is left open, and our rat (which we have seen previously in the first case of this relation) is a candidate for the argument's referent. Not every first case occupant is a referent of the argument. The rules for forming the referent subset are presented later.

While English partially segregates nouns and verbs, *gua\spi* uses the same predicate words (*qury*) in both argument and sentence phrases.

What Definitions Mean

In a dictionary words are defined in one or two sentences, but for *gua\spi* these sentences are considered to be merely a learning aid. The effective definition is a set of lists of thus-related referents. For example, the referent set of *"eats"* includes a member list with our example rat in first case and our example cheese in second, as well as numerous other members containing other rats, foods, and so on ad (almost literally) infinitum. Other predicates like *"*cu-*pair"* have referent sets that are actually infinite.

Language users are not expected to be familiar with every object list that was, is now or ever shall be thus related. A big part of language behavior consists of the listener adding to his knowledge of which items are thus related, which information the speaker sends to him or her. Each person has his own limited experience of the world, but we speak of *"the referent set"* of a word independent of a person because words are supposed to mean the same thing to each person, and language users really do agree on meanings most of the time.

Humans are very good at generalizing from a few referent set members so as to recognize novel referents, and they are not satisfied with a word until they can do such a general recognition algorithm and usually come out with the same answers their neighbors do. But mechanical users of *gua\spi* cannot be expected to show such skill, and neither can beginning human users such as infants. They must build up a referent set for a word by exhaustively hearing referent set members. If an advanced human, or advanced software, can transcend the official definition of *gua\spi* words, that's fine — a common (but risky) strategy for humans will be to use their native language as a guide to *gua\spi* meanings. However, *gua\spi* words are still defined officially in terms of referent sets simply because this definition is known to be tractable both for theory and for practical implementation. A *gua\spi* referent set is perfectly suited to be represented as a Prolog database, if truncated to a practical size.

The Interpretation of Language Behavior

When you speak an argument in a nonsentence you call the listener's attention to its referents. For example,

> **^:i |va -jiw /vn -sper -jiol** *Hey, a crocodile!*

When you speak a sentence or a subordinate assertion you do the same thing: you call the listener's attention to the members of its referent set. (Thanks to John Parks-Clifford, editor of *The Loglanist*, for this insight [TL43].) Thus in:

> **^:i |qnu !qo -jan /tara /jun !kseo |zey !ju** *John, the rat is after your cheese!*

your knowledge of the referent set of *"*jun-*hunt"* includes a member which John will want to append to the ones he knows, before the cheese is stolen. This is the ultimate meaning of the *gua\spi* sentence.

The Organizational Syntax Level

Now that we have an unambiguous parse tree made up of phrases, what shall we do with it? Modern theories of parsing are very good at describing the transition from input tokens to a parse tree, but they leave the subsequent use of that structure pretty much to ad-hoc patchwork, and *gua\spi* is no better. However, the use of the parse tree can be divided profitably into two phases, organization and semantics. Semantics in *gua\spi* consists mainly of computing and updating referent sets, whereas organization refers to a collection of preparatory transformations including assigning sub-phrases to cases, handling imbedded sentences, replacing pronouns by their antecedents, and transforming compound words into sub-phrases. One way to look at the organizational level is that these surface structures of -gua!spi are transformed into a single deep structure, the predicate with its arguments, which is a uniform and simple interface into the semantic level.

Which Words Go in Which Cases

The tones of grammar deliver to the organizational syntax level, for each phrase, an ordered list of attached sub-phrases, which are the arguments of the phrase predicate. For example in **!tara /crw !kseo**, *"*tara-*rat"* and *"*kseo-*cheese"* are attached to *"*crw-*eats"* as sub-phrases and therefore are its arguments. In the simplest and most common variation the arguments fill a sentence predicate's cases in order by number, much like English and Chinese, so *"*tara-*rat"* fills the first case of *"*crw-*eats"* and *"*kseo-*cheese"* fills the second. In arguments the first case is skipped over, being filled by an invisible placeholder for the referent. This organizational syntax can be so simple because the grammar delivers unambiguous lists of arguments, whereas in English or Latin a combined syntax has to deal with both getting the arguments on the right predicates and getting them into the right cases, and so is a lot more complicated.

The root phrase is assumed, in the absence of special cue words like **vn**, to be a sentence; thus its first sub-phrase fills its first case. All sub-phrases are assumed to be arguments with empty first cases, except if they have tones or prefixed cue words that make them subordinate or infinitive clauses.

Should it be inconvenient to have cases filled in order, *gua\spi* has ways to change the order. First, certain prefixes signify that the relation word is *"converted"*: one case is moved in front of the others. This is most useful for arguments, and lets one *gua\spi* word do the job of as many as five English words. For example in **zu -crw** the second case comes first, and the referent of such an argument would be something occupying the second case of *"eats"* before conversion: the meaning is *"food"*. The original first case, the eater, comes afterward: **zu -crw !xo -tara** means *"rat food"*. The English *"passive voice"* is a conversion in a sentence.

Second, an argument can be directed to a specific case by a *"caselink"* prefix analogous to the case endings of Latin. For example, take

^**:i !tara /fer !se -dowu** *The rat carries (something) from the house*

fer means *"X1 carries X2 to X3 from X4 via X5"*. Its arguments are **!tara** in the first case, but **se** links the next argument, "dowu-*house*", to the fourth case: the start point.

A predicate word can act as a caselink too, linking a *"modal case"* by means of an imbedded sentence. Such sentences are covered in the next two sections.

The motion words have complicated definitions, so all the definitions have been made similar: *"X1 (moves) to X2 from X3 via X4"* or *"X1 makes X2 (move) to X3 from X4 via X5"*. Many other word categories have uniform definitions too.

Sentences as Arguments — Infinitives

A *gua\spi* sentence or argument expresses a relation between specific referents, and this specific referent set member is called an *"event"*. (Frequently the sentence represents several similar events.) It is common for several cases of the predicate to be vacant: in the previous example the thing carried, the destination and the route were not specified. Nonetheless there must have been a thing carried, a destination and a route, and the sentence asserts a relation between all five arguments. The next organizational elements we will look at are linking words that attach sentence predicates (with their arguments). The linked sentences represent lists of specific events with specific argument referents and with all cases filled even if not specified by words.

Returning to organization, the first sentence link word is **vo**, which acts to change a sentence into a one-argument predicate, referred to as an *"infinitive"*, which means that the occupant of its first case is an instance of the sentence relation. Though the infinitive can itself be a sentence predicate it is much more commonly used in arguments, like this:

^**:i !tara /vyu !vo -crw !tara** *The rat likes for the rat to eat*

vyu means *"X1 enjoys doing (*vo*) X2"*, where the second case is some kind of activity — a natural place to fill with an infinitive. The sentence linked by **vo** is **!tara /crw** = *"the rat eats"*, and an instance of that relation, an event, is the referent of the argument **vo -crw !tara**. (The argument **!tara** may come before or after the sentence predicate **crw**, wherever convenient.)

vyu includes the prefix **vo** on its second case by default, as do all words which commonly have infinitive arguments. Also, there are various patterns in which main sentence arguments are replicated into infinitives, and by far the most common is for the argument just before the infinitive to be replicated, if the infinitive has none — **!tara** here. So you could just say

^**:i !tara /vyu !crw** *The rat likes to eat*

Subordinate Clauses

A subordinate clause is a sentence within a sentence. Its predicate relates one (or more) of its internal arguments to the phrase it modifies; the internal argument is called a *"modal case"* of the modified phrase. A subordinate clause can specify a tense, location, possession (genitive case), listener (vocative case), speaker in dialogue, gender, plural number, repeated action, and numerous miscellaneous cases as in the examples below. Its purpose is signalled by a linking word:

ve

A supplementary comment, giving additional information about the modified phrase, typically adding a modal case.

vu
A restrictive clause, which events of the modified phrase must satisfy, or they are thrown out of the referent set.

vi
A discursive comment, a helpful assertion by the speaker of the relation between the modified phrase and the previous sentence.

va
A supplementary assertion often stating the speaker's relation to the modified phrase.

Subordinate clauses are more common in *gua\spi* than in English, and also can be complicated, so two special rules are provided to make them simpler:

- Because subordinate clauses are so common the tone | is allocated specifically to them which automatically supplies the linking prefix **ve**. When this tone does not apply, of course, **ve** may be used explicitly. | is used commonly with the other linking words too.
- The restricted phrase is automatically replicated via a placeholder pronoun into whichever case of the subordinate clause is intended for an event, indicated by default **vo** or **bi**, or the first case if no event is expected or if the event case is occupied.

Here is a subordinate clause restricting an argument, illustrating **vu**:

> **^:i !tara /crw !xi -kseo |vu -xel** *The rat eats smelly cheese*

Not all cheeses but only those which smell (*/xel*) are eaten by the rat. The restricting sentence is *"X1 smells like X2"*, and argument referents (cheeses) are automatically brought into X1 through the placeholder. When in English we use adjectives and adverbs, in *gua\spi* we usually use subordinate clauses like this one. Here are examples of the other subordinate linking words:

> **^:i !tara /crw !kseo ^ve -tum !vden !xgno** *The rat eats the cheese with its teeth*

This subordinate clause adds a modal case. The clause is **!vden !xdro !fu -tum !vo (event)** = *"its teeth are the tool for doing (event)"*, and the asserted relation **!tara /crw !kseo** = *"The rat eats the cheese"* also satisfies this sub-sentence. Because of the clause we know that the rat does not gum the cheese. The effect is as if an additional case were added to "crw-*eat*" for the cutting tool.

> **^:i |vi -bwy ^tara |va -cul !zu -crw /fi -go -crw !kseo** *But the rat, which was full of food, didn't eat the cheese*

|vi -bwy appears to have no modal argument, but for this discursive category a pronoun is provided by default to represent the previous sentence. Thus the subordinate clause says *"this sentence differs from the previous one"*. The other clause beginning with **va** is a subordinate assertion, which is similar to a main sentence, but the reader can understand it better when it is imbedded.

> **^:i |va -tan /pur -far !tara |zey !ji** *Damn, my rat ran away*

In "tan-*annoy*" who is annoyed? "ji-*me*" is provided by default in the first case (after conversion) of supplementary assertions, main phrases and infinitives that otherwise lack one (provided certain conditions are met). Forceful or

emotional speech seems more free and expressive with this feature. The other clause **|zey !ji** is a possessive phrase; most languages have special grammar just for possessives, but *gua\spi* uses the general clause mechanism.

Pronouns Represent Words, Not Things

The next organizational issue is the pronoun. English pronouns have referents just like any other argument. But *gua\spi* pronouns represent words, not the referent of words. In computer terms, they are like functions that are expanded in-line rather than being called. The represented words are called the *"antecedent"* of the pronoun, and the sentence is analysed as if each pronoun were taken out and replaced by its antecedent. The antecedents, not the pronouns, have referents. In this way the organizational syntax level can be kept free of meaning, and the semantic level has to deal with only one class of words, predicates.

For example, a document typically will have a signature line saying in effect *"this text is the output of Jim Carter"*. (Spoken discourse is analogous though identification is by sight or voice tone.) Then when there appears the pronoun **ji** (*"me"* in English) the effect is as if the words *"Jim Carter"* had been written in its place. That is, *"A rat ate my cheese"* and *"A rat ate Jim Carter's cheese"* mean exactly the same thing.

As illustrated below, various kinds of context are carried into a phrase by a pronoun-like mechanism. When an antecedent is replicated to replace a pronoun the context is replicated with it, so the antecedent will have the same referent in both places despite intervening context changes. And when the antecedent is copied any pronouns originally in it have already been replaced by their own antecedents.

Gua\spi includes question pronouns, phrase-relative pronouns, names and modal pronouns. For question pronouns the listener is supposed to say the antecedent; in other words, the speaker provides a sentence and the listener is to fill in the blanks. There are question pronouns corresponding to English *"who"*, *"how"*, *"how many"*, *"which"* and *"isn't it"*.

Phrase-relative pronouns are for copying neighboring phrases — arguments or entire sentences. One of the more common phrase-relative pronouns is *vgry*, the whole phrase that the current listener last said, which typically is the question the speaker is filling blanks in.

In *gua\spi* a name is a pronoun. A name consists of a predicate prefixed by **qu**, or **qo** for foreign names, which disconnects the usual meaning of the predicate and substitutes the pronoun behavior. People are assigned permanent names at birth through a performative (ritual) statement like this:

> **^:i !qo -ben /ga -zu-xim !jw |cil** *Ben is (declared to be) the name of this child*

From then on, **!jw |cil** (*"this child"*, with context so the listeners remember which one) is the antecedent of the name **qo -ben**.

The six variables *da, de, di, do, du, dy* are names which you can assign to important concepts in nonfiction or characters in fiction. In mathematics it is also common to use letter words as pronouns for mathematical expressions.

Modal pronouns are like *"*ji*-me"* and *"*jn*-now"*. A modal pronoun's antecedent is set by a modal phrase with a special prefix, saying to save the modal phrase on a kind of stack, separate for each modal predicate, from which it can be retrieved. The previous antecedent can be replaced, but of more interest, it can be saved and later restored.

The modal stack is used for more than modal pronouns, though. For each kind of modal case, e.g. tense or speaker, every sentence that lacks a modal phrase for that case gets the stacked phrase automatically. Here is an example of stacked speaker cases, in story dialog:

> **^:i |qe -jai !qo -kira /py /zu-zni !cyr -far !ju** *Says Kira, Why do you flee? (default saved, set)*

> **^:i -po -sfa -daw -can -siw -dan !ju** *Don't you want to be rescued? (default inserted automatically)*

> **^:i |va -pli ^vi -zu -gre ^jo /kuo !ji** *Please, at least talk to me! (default inserted automatically)*

> **^:i |qa -jai ^qo -kira /jun !suy** *Kira pursues the swimmer. (prior default = narrator)*

Kira said the first three sentences. In the first, **|qe** indicates that the current speaker, the narrator, should be saved while Kira speaks. The words **|jai !qo -kira** are added to the second and third sentences automatically. Finally, **qa** restores the narrator as speaker and his modal phrase is put on automatically.

Tenses are handled the same way. If you put **|qe -cnu !X** on the opening descriptive sentence (where X is an event identifying when the sentence happens) then it will be propagated to subsequent sentences automatically — unlike in English where a syntactically complicated and less precise tense has to be used on every sentence. John Parks-Clifford, then with the Loglan Institute, originally developed this concept of tense defaults [TL43].

Compound Predicates

A key organizational element of *gua\spi* is the compound predicate, a sequence of words heading a phrase. The motivation to make compounds is threefold. First, you can use a single argument list to say what amounts to two sentences, which when compounded are much easier for the listener to interpret. Second, just as we use Latin prefixes in English to make many words from one, e.g. *"ob-ject"*, *"pro-ject"*, *"in-ject"*, *"ab-ject"*, most meanings in *gua\spi* are achieved by combining a much broader range of predicates. A beginner can learn the primitive words (*qury*), about 1400, and then stick them together in self-created compounds which he can expect any listener to understand, while to achieve the same range of expression in natural languages the speaker and the listener must master a huge vocabulary in which most of the words are rarely used. Third, the compound words are deconstructed into phrases headed by one *qury* word each, and semantically processing these phrases is much easier than in natural languages because there are so few *qury* that must be kept track of.

There are three main patterns to the compounds. First, if the main word has a case with a default linker of **vo** or **bi** — that is, a case for an infinitive — a word compounded with high even tone - is the predicate of that infinitive, and the main word case before the infinitive (before conversion) becomes the infinitive's first case. (Exceptions are noted in the dictionary.)

^:i !qo -kira /can -xna !fyni			Kira takes hold of the oar
	can	X1 changes so (*vo*) X2 becomes true	
	xna	X1 holds X2 with (body part) X3	
	!qo -kira /xna !fyni		Kira holds the oar (X2 of */can*)

Second, the words may share an argument list. The effect is as if you had made two sentences with the arguments copied into each. This pattern is cued by tone - when the infinitive argument pattern does not apply, or by a conjunction **-fe** when it does.

> **^:i !do /suy -pne -qmy !kqua** *It swims down through the water*

> **!do /suy !kqua + !do /pne !kqua + !do /qmy !kqua** *It swims to water; it penetrates that water; it is above that water.*

A third pattern is found in which a transitive main word is followed by its object as a compound. It is cued by the tone =.

	^:i -spo !bri =kqua |bir ^dri =fli		Maybe the pilot already drowned	
		bri	X1 breathes X2	
		kqua	X1 is a serving/portion of water	
		bri =kqua	X1 drowns	
		dri	X1 drives X2 to X3 . . . (transitive motion word)	
		fli	X1 flies to X2 . . . (motion word)	
		dri =fli	X1 pilots the flyer (airplane) to X2 . . .	

Though humans like to think of compound predicates as separate words analogous to the primitive words, compounds are actually defined through these transformations, so that each primitive word heads a separate phrase. For example in the third type of compound the compounded object is to be taken off and put in its proper case as a sub-phrase. Thus one can easily and reliably interpret a compound word that one has never heard before, as long as one knows all the primitive words.

That is how *gua\spi* is organized. Let us now turn to the semantics of arguments.

Argument Semantics — Referent Sets

As stated earlier, a predicate word expresses a relation between the occupants of its cases, and is defined by a referent set consisting of lists of case occupants that are thus related.

To interpret an argument, you start with its predicate's referent set. You retain members consistent with any sub-phrases. From each member you extract the first case occupant (after conversion), and out of these you make the *"full referent set"* of the argument. The *"referent subset"*, which is the set of actual referents of the argument, is a subset of the full set which depends on a prefix word called an *"article"* (*tirl*). (More modern terminology might be *"determiner"*.)

The most common article is **xe**, and it is assumed with most predicates when arguments lack an article. Its English translation is *"the"*. The referent subset is whichever members the speaker has in mind to talk about, but generally there are prior context cues to show which out of numerous possibilities are intended as the referents. In particular, if a set of referents has been designated before and if it is the only such set that is a subset of the full referent set of the argument, then those are the referents of the argument. For example,

> **^:i !xo -fkar |xda ^vu -xge /fi -can -tai !qel =fkar** *An old, black car emerged from its garage.*

> **^:i !fkar /cyr -vle** *The car turned left.*

*"*fkar-*car"* appears three times. The first instance designates one referent in detail using **xo**, described below. The other two instances are typical arguments with *"*xe-*the"*, but the article is unseen, being provided by default. Since the prior referent fits this predicate (and in the second sentence *"its garage"* does not), the prior referent is being redesignated. Because *gua\spi* words are so short it is just as efficient to redesignate an argument like this as to use a phrase-relative pronoun, so pronouns are less commonly used in *gua\spi* than in English.

There are also articles that select the entire referent set, *"typical"* members of it, and no members of it (actually making a negative statement about all members). Another important article is **xo**: From the full referent set one or more members are selected, and it doesn't matter which ones. For example,

> **^:i |vi -pli ^jo ^sa -ji /gey !xo -kliw** *Please give me some nails*

All in the box are equivalent and it doesn't matter which you get. **xo** is often used for arguments in the *"serving or portion"* category, called *"partitive nouns"* in English.

There are two articles for each meaning; the first unfolds the referent subset so each member is a referent, while the second specifies that the referent is the referent subset as a set. The careful distinction between sets and extensions of their members is characteristic of *gua\spi.*

Most Indo-European languages distinguish between genders and numbers of arguments. Like Chinese and English, *gua\spi* has no gender, though you may use a subordinate clause like *"*|fmy-*female"*. Number comes from the referent sets, not the grammar. You may specify the exact number of referents with a numeric predicate, like this:

> **^:i |vi -pli ^jo ^sa -ji /gey !xo -beol |fmy ^vu -zu -cu** *Please give me two nuts (female screws)*

Vocabulary

A great deal of the machinery of language, which in natural languages is shared between the grammar and the vocabulary, is handled in *gua\spi* purely by words. Here is a discussion of how the words were created, and four samples from a long list of models of how to say things. Frequently I have thought that some form or meaning required a new primitive word, or even a change in the grammar, but it has turned out that existing words were more than adequate if creatively used.

Word Creation

The words of natural languages appear to be arbitrary symbol strings of tremendous variety of sound. *Gua\spi* is similar in that its words were generated by a partially random process. However, the words were made to resemble natural language words so they would sound more pleasing and so occasional cognate relations might aid learning. To begin, the word lists of *Loglan* [L4] and *Lojban* [Lja] were merged and some additional words were added. An English, Chinese and

Latin translation was determined for almost all words. (Latinoid English words were avoided.) Both *Loglan* and *Lojban* use many more natural languages as word creation fodder.

Then experimental phonetic data [NB2] was used to assess candidate words for the accuracy with which speakers could distinguish them. For each *gua\spi* meaning, randomly generated word candidates were evaluated for recognizability, for distance from other *gua\spi* words, and also for similarity to their natural language equivalents. The final assignments were determined through a process of numerical annealing so as to maximize the summed quality scores.

CV structure words were assigned by hand; related structure words, like articles, have the same C and varying V's. Structure words pertaining to numbered cases have the same V's as the corresponding digits, but contrasting consonants, making learning easier.

A question often asked is, why create new words? Why not use Chinese or English words? First, some attempt has been made to keep *gua\spi* culturally neutral, and if Chinese words were used it would intimidate English speakers and vice versa. More important, Chinese words are designed for use with Chinese. Many required meanings, like articles, simply do not exist in Chinese, and those meanings that are present are only approximations of the *gua\spi* meanings. That is why the approach was rejected of simply stealing natural language vocabulary.

How adequate is the word list? Can every required meaning be expressed by infinitive, parallel and transitive compounds? Only extensive literature in *gua\spi* by a variety of authors can demonstrate adequacy. However, I have written about 20,000 words of prose and fiction in *Loglan*, and I am satisfied with the coverage of the *Loglan* word list, especially with the *Lojban* additions and with my own. Some people are interested to discover just how few basis words we can get by with. While I do not believe that the *Lojban* word list is minimal, I think it is fairly close. Thus I chose to use existing word lists for *gua\spi* rather than to try for radical pruning or *de novo* creation.

Modal Cases

Here are examples of a few modal cases. However, virtually any word can be construed as a modal operator. Be alert for creative opportunities for expression.

bir

Past tense

^:i !ji /crw !kseo ^vu -bir !jun -vnl !tara *I ate the cheese before the rat came hunting*

zey

Genitive or possessive case: a relation of pertinence

-fkar |zey !ji *My car (which I lease and which my brother is now driving)*

jai

The speaker

^:i |jai !qo -kira /ju /dwu -csn -zu -jeu *Said Kira, You're a monster*

plm

Such as: an example

-xy -pso |psi ^vu -plm !xy -kai |kei *Bad people such as thieves*

Varieties of Negation

In *gua\spi* negation is not a unitary concept; beside the obvious antonyms there are nine or ten ways to express negative meanings, most of which involve compound words. Here are a few examples.

^:i -go !ji /kio !tara |zey !ju
I *don't* have your rat. **go** is a mood prefix which means that the asserted sentence is counter to fact.

^:i !jw |kseo /fi -gl -zao
This cheese is flavor*less*. Some dimensions like *"*zao-*flavor"* are quantifiable (more or less) but unsigned, so their degree ranges from zero to larger values. Others like *"*gal-*high"* are signed. In either case **gl** modifies a predicate so that its degree is zero or negligible.

^:i !jw |kseo /fi -gr -ksi
This cheese is *not* fresh. When the dimension ranges from positive to negative values, **gr** interchanges positive and negative. For extremes of unfreshness one can use *"*fpu-*rotten"*.

^:i !jw |kseo /fi -vry -can -psl
This cheese is *de*solidifying. *"*vry-*reverse"* indicates that the process in its X2 case is occuring in the reverse of the usual order.

Causal Connectives

The root structure of syntax is a discourse, or sequence of sentences. But the sentences need not stand alone; they may be connected by predicates, like this:

^:i -dae !kara ^:o -bal !crw |jro ^tara ^kseo *If the box is open then maybe the rat will eat the cheese.*

The speaker may connect sentences with any useful word having suitable cases, such as *"*kau-*cause"*, *"*kmo-*motivate"* or *"*sny-*imply"*. Like all *gua\spi* words, the causal connectives can also be useful as arguments and as modal caselinks. In this example **^:o** is a *"retroactive downjump"*, a special case in organization. A sentence start word, it transforms itself into a pronoun for the previous sentence, which goes into the first case of the following main word, the causal connective in these examples. Human speakers prefer infix causal connectives with a retroactive downjump rather than the obvious form with two explicit infinitives.

Mathematical Expressions

Any discussion sooner or later involves quantitative statements with units of measure. Therefore *gua\spi* has quite an extensive facility for mathematical expressions, even if the more complicated possibilities are rarely used by non-scientists. First, *gua\spi* syntax matches perfectly the definition of a *"number"* as an equivalence class of equal-count sets. This concept can be generalized to various extension rings and fields.

> **!xu -cu -cw -ci** *The number 2.5 (the class of all sets of count 2.5)*

Mathematical functions are defined with such classes as formal parameters, and hence have **xu** on parameter cases by default — **xu** means the entire referent set of an argument, as a set (or class). The first case of a function is its value, and the function is defined as *"X1 is in the equivalence class that comes from doing (function) on (*xu*) X2"*, possibly with several parameters. For example,

> **^:i !xa -ca /plw !co ^cu** *3 is the sum of 1 and 2 (every triplet is in the equivalence class of 1+2)*

Specifically, units of measure are defined to multiply a number or other expression by the unit. The resulting equivalence class is considered to contain *gua\spi* events whose degree or measure are that big; hence the unit expression takes the form of a subordinate clause, and the main sentence predicate tells what dimension is being measured. For example,

> **^:i !ji /vga |kyam !ku -cy** *I weigh 70 kilos (I heavy kilo 7 0)*

Sample Text

Here is a short passage from a story I am currently translating from *Loglan* to *gua\spi*. If a second human knew *gua\spi* I would have him or her translate it to English, but so far my only colleague is mechanical. Its purpose is to check syntax and organization, and its English is only good enough to substitute for a full parse tree printout. For example, it can't tell an infinitive from a gerund, and it is overly free with possessives. Nonetheless you can get an idea how much meaning a mechanical translator can recover from the text.

Brackets `[ ]' surround sentence-type phrases; angle brackets `<>' mark subordinate clauses; backslashes ** repeat the predicate of the phrase a subordinate clause restricts; parentheses `()' enclose the antecedent of any other pronoun; and when a word's meaning as an argument differs from its root meaning, the root comes afterward in slashes `//'. Subscripts give the case number of each argument.

> **^:a |vi-gza ^vu-qe-kam !sa-cil ^qu -jaiw=tiri |va-ga-xim !do /fi-jiw !su-cana-fer ^vi-gau ¶** *Tigereye cries to the children, the barge, look out!*

[then the boat$_1$ and$_2$ carry$_2$ surprise < paragraph \surprise$_1$\ < speaker/cry/ tiger's$_2$ eye$_1$ < performative name \eye$_1$\ variable b$_2$> \surprise$_2$\ < listener/warn/ you$_2$ (child) \surprise$_2$\ < time/present/ (something$_2$) \surprise$_1$\]

"Carry-boat" is used for *"barge"*. **qe** sets the speaker and listener modal case antecedents. However, *"*gau-*warn"* supercedes *"*kam-*cry"* as the predicate bearing the listener. *"*do-*variable~b"* is assigned to represent Tigereye. Words for document structure like *"*gza-*paragraph"* fit naturally into the text, written or spoken.

> **^:i |faw ^jo /suy !jr** *Swim over here!*

[imperative$_1$ (child) swim place$_2$ (something) < listener/emphatic/ you$_3$\(child) \swim$_2$\ < speaker/emphatic/ I$_1$ (eye) \swim$_2$\ < time/present/ (something$_2$) \swim$_1$\]

jo instead of *"*ju-*you"* makes the sentence imperative. *"*faw-*emphatic"* has a default *"*ji-*me"* in the case for its speaker and *"*ju-*you"* in the case for its listener, thus superceding the default *kam* otherwise imported from the first sentence.

> **^:i |qi-qnu !qo-qosefo /ve-faw ^jo /qma -duw !gunu !ju** *Josepho, move your ass!*

[imperative$_1$ (qo se fo) make to$_2$ [your$_2$ (qo se fo's) buttock$_1$ move] < listener/emphatic/ you$_3$ (qo se fo) \make$_2$\ < speaker/emphatic/ I$_1$ (eye) \make$_2$\ < time/present/ (something$_2$) \make$_1$\]

Only a few words like *"*fer-*carry"* are intrinsically transitive. Normally an infinitive compound with *"*qma-*make"* makes words transitive.

> **^:i -po !ju /daw -scu-zu-crw-zu-ter !fwa-pei !cana-fer** *Do you want to be chewed up by the barge's propeller?*

[you$_1$ (qo se fo) is it desire to$_2$ [to$_1$ [[(qo se fo$_2$) food$_1$/eat/ by to's$_2$ [push boat$_1$ and$_2$ carry$_2$] device$_1$] and [(device$_1$) tear$_2$ (qo se fo$_2$)] complete] < speaker/cry/ (eye$_1$) \desire$_2$\ < experiencer/attention/ (qo se fo$_1$) \desire$_2$\ < time/present/ (something$_2$) \desire$_1$\]]

A complicated sentence, just as efficient as English. See how Josepho is replicated into the internal infinitives. *"Push-device"* is used for *"propeller"*; *"completely eat-tear"* is *"chew up"*. The speaker and time are added automatically.

> **^:i |qi-koy !do /va-gri ^ji /gul !qou !jw ^vu-cnu !xa-jl** *Damn, I have to watch them every moment!*

[I$_1$ (eye) must to$_2$ [(eye$_1$) watch$_2$ this$_2$ < time/present/ all something$_2$ \watch$_1$\] < performer/think/ variable b$_1$ (eye) \must$_2$\ < listener/think/ (eye$_3$) \must$_2$\ < actor/angry/ (eye$_2$) I$_1$ (eye) \must$_3$\ < time/present/ (something$_2$) \must$_1$\]

A subordinate clause gives, literally, *"at all times"*.

> **^:i !pu /kau !qai-kar !jw ^jw /vi-faw** *Why can't they take care of themselves?*

[what$_1$ cause to$_2$ [this$_1$ fail to$_2$ [(this$_1$) care$_2$ this$_2$]] < listener/emphatic/ you$_3$ (eye) \cause$_2$\ < speaker/emphatic/ I$_1$ (eye) \cause$_2$\ < time/present/ (something$_2$) \cause$_1$\]

For a reflexive, repeat the argument.

> **^:i !ji /suy |swa !dman =co** *I swim for one second...*

[I_1 (eye) swim < duration all* one's$_2$ second$_1$ \swim$_2$\ < performer/think/ variable b_1 (eye) \swim$_2$\ < listener/think/ (eye$_3$) (swim$_2$)> < time/present/ (something$_2$) \swim$_1$\]

Here is another tense, this time a continuous one. *"One second"* is a simple mathematical expression.

> ^**:o -sno !can-zu-vem !jw** *And that's enough for them to get into trouble.*

[conjunction \swim$_1$\ sufficient to$_2$ [to$_2$ [(this$_2$) trouble] change by this$_1$] < performer/think/ variable b_1 (eye) (sufficient$_2$)> < listener/think/ (eye$_3$) (sufficient$_2$)> < time/present/ (something$_2$) \sufficient$_1$\]

Gua\spi is much richer in causal connectives than English with its ambiguous *"because"*; thus this *gua\spi* sentence is half the length of its English translation.

This short sample exercises almost every feature of *gua\spi*\ even including a mathematical expression. To verify every phrase the reader must know *gua\spi* fairly well, but one can see easily the simple phrase organization of *gua\spi*. The lengths of the *gua\spi* and native English sentences are comparable, showing how efficient *gua\spi* is. The primitive words cover almost all meanings in this relatively unspecialized text, and the componds for *"barge"* and *"propeller"* are quite understandable. Finally, the sample makes it clear that *gua\spi* is more than just a dry substitute for SQL; *gua\spi* can support real life.

Conclusion

What can one use *gua\spi* for? Here is a brief list:

- In knowledge representation one always wonders if one's codes can really cover the totality of human experience. The *gua\spi* word list has some history from which general coverage can be claimed.
- *Gua\spi* provides a medium in which real humans can have real conversations, after which you are guaranteed that a computer can semantically analyse what was said so that you can study it. Freely spoken natural languages are too complex to handle.
- The phonology and morphology of *gua\spi* is simpler than that of typical natural languages, so *gua\spi* could be useful as a test case for automatic speech recognition, especially if semantic analysis of arbitrary speech is a goal.
- The modes of thought you typically use in *gua\spi* are much more precise than is typical in English or other natural languages, and the connections between sentence parts are much clearer. *Gua\spi* can be a kind of mirror of your native language that can help you use it better, and the *gua\spi* thought patterns may affect the content of your thoughts — the Whorf-Sapir hypothesis.
- *Gua\spi* sidesteps the recognition algorithms by which normal humans assign objects to referent sets. A research project to incorporate such algorithms in a *gua\spi* semantic analyser would be very interesting.
- Another interesting project is to investigate how small the primitive word basis can be.
- The simplicity and precision of *gua\spi* helps a learner even if he is not mathematically inclined. *Gua\spi* would make a good substitute for pidgin English in linguistically impoverished areas.

I hope this brief introduction to *gua\spi* has whetted your appetite to learn more about it. As you have seen, it expresses typical human sentences easily and efficiently. But the meanings of the words, and particularly the meanings of the phrases and sentences made from them, are defined much more specifically and clearly than in even the best natural languages. Finally, and most significant for artifical intelligences, the resulting meanings are cast in a form that is ideal for modern fifth-generation languages — which, in fact, those languages were designed to represent. Thus the gap between human and machine languages is closed by *gua\spi*.

Appendix:

Gua!spi Grammar in Backus-Naur Form *"Discourse"* is the root grameme. Grammar for quoted non-*gua\spi* text is not shown, but foreign predicates and quoted *gua\spi* are processed by this grammar and are recognized at the organizational syntax level. A procedural definition shows the simplicity of the grammar more clearly.

; Morphology.	
C	= (choice of letters)
Cseq	= (Cseq C) | C
V	= (choice of letters)
Vseq	= (Vseq V) | V
Word	= Cseq Vseq
; Tone categories.	
Compound	= -' | ='
Sametone	= `^'
Down1	= !' | |'
Up1	= `/'
; Grammar. LHS `-' symbol indicates which end has a tone.	
Prefix	= (subset of Word, e.g. *vo''</I> or <I>zu'*)
Primitive	= (subset of Word, e.g. *tara''</I> or <I>crw'*)
Phrase	= Prefix Args0 Phrase
<TD> | Phrase-w	
Phrase-w	= Primitive Phrase-w
<TD> | Primitive	
Phrase0-	= Phrase Sametone
<TD> | Phrase Down1 Args1	
; Args(n) is a list of phrases that jumps up n levels at the end. Args3, 4, . . . are defined similar to Args1 and 2. Some finite bound must be set on n to give a finite grammar.	
-Args0-	= Compound (Just one tone)
<TD> | Down1 Args1	
Args1-	= Phrase Up1
<TD> | Phrase Down1 Args2	
<TD>	

Bibliography

- [L1] Brown, James C. Loglan 1: A Logical Language. The Loglan Institute, Inc., Gainesville, Fla., 1975.
- [L4] Brown, James C. Loglan 4 5: A Loglan-English / English-Loglan Dictionary. The Loglan Institute, Inc., Gainesville, Fla., 1975.
- [NB2] Brown, James C. A Proposed Revision in the Structure of Loglan Words (Notebook No. 2). The Loglan Institute, Inc., Gainesville, Fla., 1982.
- [TL43] Parks-Clifford, J. Supplement to Loglan 1. \sl The Loglanist 4, 3 (Nov. 1980).
- [La] Information about *Loglan* may be obtained from The Loglan Institute, Inc., 1701 NE 75th Street, Gainesville, FL 32601.
- [Lja] A modernized version of *Loglan*, much closer to the original *Loglan* than *gua\spi*, is *Lojban*. Information is available from The Logical Languages Group, 2904 Beau Lane, Fairfax, VA 22031, or lojbab@lojban.org. For on-line access, send a message whose body is *"index lojban"* to listserv@hebrew.cc.columbia.edu.
- [Ga] Please contact the author at UCLA Department of Mathematics, Los Angeles, CA 90024-1555, or e-mail to jimc@math.ucla.edu.

Gua\spi Reference Manual

James F. Carter

20 February 1989

Abstract:

Gua\spi

is an artificial language suited to both humans and machines. It can express real human conversation. Yet the vocabulary and the grammar are two and three orders of magnitude simpler than English. Word and phrase meanings are defined through predicate calculus and hence can be represented and manipulated efficiently and unambiguously by programs (and people).

Introduction to *Gua\spi*

Most human languages are natural: they evolved with their host societies without the benefit of intentional design. While some languages like French are maintained by dedicated stewards, and others like Spanish and Russian have been renovated recently, most drift with the fashions of peasants and teenagers which, though vital, lack logic and efficiency. A very few languages, however, have been created as artifacts with specific goals in mind. *Gua\spi* is one of these. My goals in building *gua\spi* were:

- To investigate the nature of language, and particularly the minimum content required for a language, through engineering and experiment.
- To create a language suited to use by artificial intelligences, such that the effort to map from letters to meanings does not overshadow the effort spent on using the resulting meanings.
- To create a language for my own use, free of the limitations of English, and to have fun doing so.

The purpose of this monograph is to present the syntax of *gua\spi*, as well as categorical information about the vocabulary, in the style of a reference manual. All the syntax is here (excepting only features under active development that were too late to make publication), and the vocabulary section contains an extensive list of how to say various types of expressions. To learn *gua\spi* you also need a dictionary, a language textbook, and a set of *gua\spi* reading material.

Artificial Intelligence

The significance of *gua\spi* to artificial intelligence work is that it is a bridge between humans and machines: it is complete enough to express the real conversation of real humans to each other, unlike database representation languages, yet it is simple enough that a working Prolog parser can be put together in a few days, unlike natural languages and particularly English, which must be so stripped as to be scarcely useable before a simple parser can handle it.

Gua\spi has several characteristics that particularly suit it to use by humans and artificial intelligences together.

- *Gua\spi* is simple. The formal syntax can be stated in a few lines, compared to thousands of lines for English. There are only eight classes of structure words (occupying only two distinct syntactic sites), with about six functionally and morphologically related words per class; a similar set of pronouns; and 21 digits. The content words number only about 1400, compared to half a million for English.
- *Gua\spi* is modular. Morphology, grammar, organization and semantics are defined separately and interact to the minimum feasible extent.
- *Gua\spi* is complete. The content words form a basis such that almost any meaning not tied to a specific place or culture, and many which are, can be represented by agglutination. Foreign words and scientific Latin are welcome in the language.
- *Gua\spi* is flexible. A minimum of preconceptions are imposed on the user by the language. Trials show that *gua\spi* can express human speech from daily life as well as highly technical scientific language.
- *Gua\spi* is efficient. Words are short, and extensive defaults on articles and modal cases eliminate the majority of structure words.
- *Gua\spi* is unambiguous. There is one sound per letter and one meaning per word; and every valid utterance can be parsed in only one way.

Ancestor Languages

The language artifact *Loglan*, developed by James Cooke Brown [L1], was the inspiration for *gua\spi*. Brown realized that a very small set of content words could form a basis of a language, and produced such a set. By successfully writing large amounts of prose in *Loglan* while creating almost no additional words, I validated his insight.

Loglan was the first language to have such simple grammar — a hundred times fewer syntax rules than English, for example. But aggressive simplification can still be applied, and this I have done in *gua\spi*. One is tempted to think of the resulting syntax and morphology rules as trivial. A better way to describe them is, they contain the essentials and nothing more.

Gua\spi's syntax is much simpler than English or other languages, partly because the syntax is divided into modules each of which has its own purpose. *Gua\spi*'s syntax modules are:

Morphology
How to divide letters or sounds into words.

Grammar
Joining words into phrases into sentences.

Organization
What each phrase does in the sentence.

Semantics
Giving meaning to syntactic structures.

Natural language syntax is extremely complicated because the syntax expresses actual meanings such as tenses and numbers. In *gua\spi* the first three levels are independent of the meaning of the words. This makes them less interesting than jewels like the *"perfective aspect"* of Russian or the *"long object case"* of Navajo, but it makes them much simpler and much easier to learn and use.

Morphology: What is a Word

The phonemes are divided in two classes, C's and V's. All C's are consonants in English and those English vowels used in *gua\spi* are all in the V class, hence the names. In addition each word has a tone, a frequency modulation of the V's of each word in the Chinese manner. A word is written as a tone (see Table 4 [Tones]), one or more C's and one or more V's. What could be simpler?

The Phonemes

Phonemes can be distinguished by where the tongue is placed to make them, whether they are sudden (plosive) or continuous (spirant), and whether their sound comes from the vocal cords (voiced) or the rush of air (unvoiced). Particular ranges of tongue position produce each phoneme, much like states on a map. But each listener has unique map boundaries for recognizing phonemes, especially for the vowels, so the speaker should try to hit the center of the phoneme region so as to maximize the likelihood that any particular listener will be able to recognize his speech. Nonetheless, the more difficult phoneme distinctions have been removed from *gua\spi* and so speakers of any natural language should find most phonemes easy to say and to hear.

Table 1 [Phonemes] shows the phonemes, categorized by tongue position and sound source. Some phonemes are represented confusingly in European languages, e.g. *'sh'* which sounds like neither *'s'* nor *'h'*. So in *gua\spi* they are assigned individual letters which differ from European usage — *'q'* for *'sh'*. Table 2 [Pronunciation] gives examples of these, and all the vowels. Written blanks have no sound, and are optional. There is no distinction between upper and lower case.

C/V	Stop Class	Sound	Labial	Dental	Palatal	Velar	Glottal
C	Plosive	unvoiced	p	t	c*	k	—
C	Plosive	voiced	b	d	j	g	:*
C	Spirant	unvoiced	f	s	q*	—	—
C	Spirant	voiced	v	z	x*	—	#*
V	Vowels		u	o	y	i,e	a
V	Nasal etc.		m	n	l	w*	r

Table 1 [Phonemes]. *Gua\spi* phonemes, arranged by tongue position front to back (reading across) and sound type (reading down). Letters marked * differ from European standard usage.

Gua\spi	English	Examples of Pronunciation
c	ch	CHew, Ciao (Italian)
q	sh	SHoe
x	zh	aZure, breZHnev (Russian)
:	(pause)	the:apple, hawai:i (glottal stop)
#	uh	thE, Among (schwa)
u	u, oo	flUte, bOOt
o	o, oa	bOne, bOAt
y	i	knIt
i	i, ee	grEEn machIne (not eye)
e	e	bEd
a	a	fAther (not cAt)
mnlr	mnlr	LeMoN RiNd (no silent R)
w	ng	stroNG

Table 2 [Pronunciation]. How to pronounce *gua\spi* phonemes. Nonstandard C's are shown; C's without examples are as in English. Standard radio broadcast accent is close to correct for the V's; Spanish is closer. Pronounce the vowels as one sound, not a glide between two sounds as in *"eye"*.

The sound `#' or *'uh'* is common in English; all vowel letters are sometimes pronounced `#'. The *'a'* of *"among"* is a good example. This sound is called *"schwa"*; that German name is pronounced (with *gua\spi* letters) *"sqv#"*. `#' is not used in regular words; its purpose is to break up CC pairs that a particular speaker finds hard to pronounce, since virtually all speakers will be able to handle C#C. It is to be ignored and it is only written in explanations like this one. Though normally considered a vowel, it is in the C class because it occurs among C's, and a word is defined as some C's followed by some V's.

The glottal stop **:** pronounced alone is a sudden (plosive) `#', but it is normally followed by a V so that it sounds like a brief pause after which the V comes on. In many English dialects, as in *gua\spi*, it is found between a vowel-final and vowel-initial word, like **the:apple**, while the Cockney dialect uses it much more extensively. The glottal stop is not used in regular words; its place is at the beginning of each sentence start word, and in vowel-initial foreign words.

English has thirteen subtly different vowels plus four official diphthongs but only five letters to represent them. *Gua\spi* uses only six easily distinguished vowel sounds, recruiting Y for one of them, and adds some vowel-like sounds which are considered consonants in English. Unfortunately, many regional accents of English turn simple vowels into diphthongs, invalidating the example words given in Table 2 [Pronuncation]. Other accents transform sounds beyond the bounds that a *gua\spi* speaker can recognize. If you speak with a regional accent, please use the vowel sounds that you can hear on television or radio (American or British will both work). Particularly troublesome examples, rendered with *gua\spi* letters, are shown in this table:

Standard	Accented English	
flUte	flIUte	
bOAt	bAt	(a very closed `o')
machIne	machI#n	
bEd	bAI#d	
fAtheR	f%thA	(% represents `a' in ``cat'')

Table 3 [Accent]. Accented Vowels Troublesome in *Gua\spi*. If you normally speak a regional dialect, try to use standard pronunciation in *gua\spi*.

Japanese speakers are famous for producing *'l'* and *'r'* that Europeans cannot distinguish. Chinese has distinct *'l'* and *'r'* but uses phoneme boundaries different from the European norm, so its speakers also have some trouble being understood. The *gua\spi 'l'* and *'r'* are biased to European norms, and Asian speakers should take special care with these phonemes.

Preliminary experience shows that the errors English-speaking beginners make most often are to interchange *'q'* with *'c'*, *'x'* with *'j'*, and *'i'* with *'y'*; and to pronounce *'w'* as *'oo'* (should be *'ng'*).

Written blanks have no sound, and are optional. In this document a blank usually comes before each word (except in the phrase "gua\spi"), although in running text it looks nicer to omit blanks before the tone -. There is no distinction between upper and lower case. The tones (described next) make punctuation unnecessary. There are no periods at the ends of sentences; however, each sentence start word begins with a glottal stop, written as a colon. This colon is a letter, not a punctuation.

A feature of *gua\spi* (like Loglan before it, and unlike English) is that writing and speech are isomorphic, that is, each letter has a single phoneme (sound) and each phoneme has a single letter (with trivial exceptions), so that each spoken text can be spelled easily and without ambiguity, and each written text can be read off equally easily.

Grammar by Tones

The job of grammar is to stick words together into phrases. The grammar does not support meaning of any kind — no tenses, no possessives, no nouns, no verbs. These ideas are handled at the organizational and semantic levels, using the grammar as a foundation. Like its morphology, the grammar of *gua\spi* is nearly minimal.

Parse Tree

The grammar is stated in Backus-Naur form in Section [Backus]. For grammatical purposes there is only one kind of phrase (though distinctions are made at the organizational level), but words have five categories: the two words **fu** and **fi**, sentence start words, other prefixes, and everything else. The main part of a phrase is a sequence of one or more words collectively called the *"phrase predicate"*; any prefixes in this must come first. After any of the prefixes or after the whole predicate the sub-phrases are interspersed. They, of course, have their own prefixes, predicates and sub-phrases.

Let us understand phrases with the help of the example in the following figure, showing the *"parse tree"* of a simple sentence. The root phrase is at the top; parse trees grow upside down. Sub-phrases with their own predicates come at the next lower level. These in turn may have their own sub-phrases. Each phrase is at a certain level and it attaches to the most recent phrase at the next higher level. The tones (see Table 4 [Tones]) show the level of each word relative to the one before it.

> **^:i !tara /vme -crw !kseo ^vu -tum !kfor ^fe -fnau** *The rat devours (violently eats) the cheese with a fork and knife*

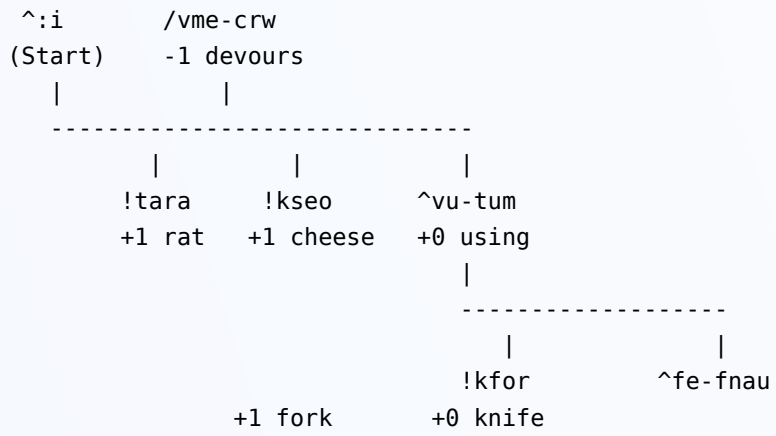

Figure 1 [Parsetree]. A Sample Parse Tree. Each phrase is part of the most recent one at the next higher level. Each phrase may have sub-phrases. Tone symbols (see Table 4 [Tones]) show jumps into and out of sub-phrases; they show the difference in level between adjacent words.

The root phrase predicate of the example sentence is *"devour"* (violently eat), and the start word is at the same level. *"Rat"*, attaching to *"violent"*, is therefore at the next level below the start word, indicated by **!**. *"Violent"* is at the level above *"rat"*, indicated by **/**. *"Eat"* is attached to *"violent"* in a compound, indicated by **-**. For grammatical purposes the pair of words, plus the start word, form the root phrase predicate, although later at the organizational syntax level *"eat"* will be split from *"violent"*. Like *"rat"*, *"cheese"* attaches below *"eat"* (**!**). The next phrase also attaches to *"eat"*, indicated by **^**, because *"eat"* is the most recent word at the next level up. It is a subordinate clause, indicated by the prefix **vu**, that tells what was *"used"* to eat; in a similar tone pattern *"fork"* and *"knife"* both attach to *"using"*. *"Using"* and *"knife"*

attach to their prefixes with compounding tone -. To summarize, words attach to the previous word at the next higher level, and the tones of *gua\spi* represent the attachment level of the present word relative to the one just before it. Every word has such a tone.

The Tones

The tones are the most terrifying aspect of *gua\spi* for speakers of European languages. Please remember that over a billion people in China and Southeast Asia speak tonal languages. If they can do it, so can you.

A tone is a specific change in pitch or vocal frequency. In English a falling tone marks the end of the sentence, a rising tone marks a question, a down-up combination is a kind of verbal comma, and most of the sentence is said at a fairly even tone. Chinese has similar tones but each word has one and the tones distinguish between meanings, e.g. **-ma** (high even tone) means *"mother"* while **|ma** (down-up) means *"horse"*. *Gua\spi* is intermediate in its use of tones: each word has a tone, but rather than affecting the meaning of the word it tells which neighboring word it attaches to. The tones are shown in this table:

Number	Sound	Level	Type	Symbols	
1	High-even	Predicate	Compound	⚙️	
2	Rising	One higher	Phrase	/	/
3	Down-up	One lower	Clause	|	⚙️
4	Falling	One lower	Argument	\	!
5	Up-down	No change	Phrase	^	@
6	Low-even	Predicate	Transitive	=	%

Table 4 [Tones]. Sounds and interpretations of the tones. *"Level"* refers to the parse tree level of the word with that tone, relative to the structure before it. *"Type"* indicates the organizational type of that word or phrase. The first set of symbols shown, ASCII characters, is preferred but the second set can substitute on a manual typewriter. In this paper, **!** is used instead of * * for convenience in typesetting.

You don't speak this language, you sing it. Chinese speakers will have no problem to produce and hear the tones, but Europeans will need a lot of practice, particularly in listening. A Chinese person speaking clearly will let his voice vary over an interval of a fourth, e.g. C to F on the piano, and this should be enough for Europeans to catch. On the falling and rising tones Europeans tend to produce, and listen for, little curlicues at the end that make the tone sound like down-up or up-down. Avoid the curlicues, and recognize the two-direction tones only if they range over at least a third, e.g. C to E or D to F.

Tones - and = join adjacent words of a compound phrase predicate. Tones | and **!** start a sub-phrase of the current phrase. Tone ^ closes the current sub-phrase and starts a new one, part of the same containing phrase. Tone / closes a sub-phrase and resumes the predicate of the containing phrase, if among its prefixes, or otherwise starts a new phrase at the higher level. Distinctions within these tone classes are important later but do not affect the grammar.

A sentence start prefix with up-down tone such as **^:i** is automatically at root level. **fi** jumps to the root level or one or two levels lower without ending the sentence, depending on its tone selected from **!**, ^ or /, as if an imaginary level 1 word preceeded it. If a word is supposed to be more than one level higher (closer to the root) than the previous one but **fi** does not apply, you use **fu** to raise the level count by two to four depending on its tone, selected from **!**, ^ or /

respectively. You can repeat **fu** for even higher levels, but this is rarely necessary since from level 7 or less you can reach any level from 0 (root) to 8 in one jump. On the level-shifting prefixes the tone - usually can substitute for ^, except after a foreign word, name or digit, where the prefix tends to get included in the previous structure.

Sentence start words like **:i** and other prefix words like **se** or **vu** come before and at the same level as the predicate of their phrase. The predicate normally has compounding (high even) tone unless some of its arguments precede it, and the linked predicate is considered to be a compound word with the prefix, whether or not arguments come between them. There may be several prefixes in a row before the predicate, all at the same level and linked in sequence. The one closest to the predicate has effect first.

As befits a grammatical element, the tone of a word implies what kind of structure is attaching. In particular, the third tone, down-up, |, is specifically for the very common case of putting a subordinate clause one level down from the previous word.

There are two word sequences for quoting non-*gua\spi* text. These are part of the grammar but are described with the other quote words, which are at the organizational level.

That is the end of the *gua\spi* grammar. A speaker used to natural languages surely will expect to have seen tenses, cases, numbers, aspects, moods and all the other baggage that inflates natural language grammars beyond the bounds of reason. *gua\spi* grammar provides a framework upon which words can mean tenses, cases and so on, but the grammar by itself eschews any meaning — so it can be much simpler than natural language.

Cases and Relations

The next layer of *gua\spi* syntax is the organizational level, but to understand the reason for some organizations we have to make a detour into semantics to find out about cases. We also get to see some examples of *gua\spi* sentences.

At the beginning we will use this sentence for an example:

> ^**:i !tara /crw !kseo** *The rat eats the cheese (start: rat eat cheese)*

Please pronounce it correctly: *'c'* as English *'ch'* and *'i'* as *'ee'*. Mind the tones, lest you change it into *"the eat rats the cheese"* or some such. Since no dictionary is included with this paper, in examples where it is hard to match up the *gua\spi* and English words the English translations are augmented with a *"pidgin"* translation using *gua\spi* word order. The notation "tara-*rat*" in examples means that **tara** is the example word, and it means *"rat"* in English. Isolated words or phrases like this are written without a tone because it depends on the context where the word is used.

What is a Predicate

Human languages generally distinguish between *"things"* and *"actions"*, where an *"action"* is a relation between *"things"*. The formal term for such a relation is a *"predicate"*. Take for example:

> ^**:i !tara /crw !kseo** *The rat eats the cheese*

"crw-*eats*", called a *"predicate word"*, tells how the rat and the cheese are related and is a symbol for a certain predicate. The predicate is like a function whose arguments are things that might be related; the value of the function is true or

false (or fuzzily in between) depending on whether or not they actually are thus related: in this sentence, whether the first actual parameter eats the second.

The formal parameters of a predicate, regarded as a function, are referred to as *"cases"*. English has *"nominative"* and *"accusative"* cases (the rat occupies the nominative case and the cheese occupies the accusative case), and Latin has in addition *"genitive"*, *"vocative"* and others, but *gua\spi* simply numbers the cases. Some *gua\spi* words have as many as five numbered cases. In our example, "tara-*rat*" fills the first case of "crw-*eats*" and "kseo-*cheese*" fills the second.

The words denoting the actual parameters of a predicate are called *"arguments"*; being sub-phrases, they have their own predicate words. Here, **!tara** and **!kseo** are the arguments. The *"thing"* represented by an argument, which is the actual parameter of the sentence predicate, is something that can fill the first case of the argument's predicate. It is referred to as the *"referent"* of the argument. For example,

> **-xe -crw !kseo** *The eater of cheese*

is an argument phrase; the first case is left open, and our rat (which we have seen previously in the first case of this relation) is a candidate for the argument's referent.

Not every first case occupant is a referent of the argument. The rules for forming the referent subset are presented later.

A predicate might have only one case. Some such words are actions in English, like "vde-*alive*", but most are things, like "tara-*rat*". By itself "tara-*rat*" is an argument, and its open first case can be filled by any one of numerous rats.

In literate English and most other languages a word should not be both a noun and a verb, but in *gua\spi* any predicate word can play either role depending on cues recognized at the organizational syntax level; the grammar is the same for sentences and arguments. This unification cuts in half the complexity of the language, which is already simple. The term *"phrase"* will be used to mean either a sentence or an argument.

What Definitions Mean

A predicate word expresses a relation between the occupants of its cases. In English and all natural languages, words are *"defined"* by a sentence or two; the words in those sentences are often defined circularly in terms of the word being defined. In *gua\spi*, on the other hand, the text definition is merely a learning aid. The relation is actually defined by a set of all thus-related object lists. For example, the referent set of *"eats"* includes a member with our example rat in first case and our example cheese in second, as well as numerous other members containing rats, foods, and so on ad (almost literally) infinitum. Other predicates (such as *"pair"*) have referent sets that are actually infinite.

Language users are not expected to be familiar with every object set that was, is now or ever shall be thus related. A big part of language behavior consists of the listener adding to his knowledge of which items are thus related, which information the speaker sends to him. Each person has his own limited experience of the world, but we speak of *"the referent set"* of a word independent of a person because words are supposed to mean the same thing to each person, that is, if a person is aware of a particular referent set member, typically he will agree with other language users which word's definition it is a member of.

Humans are very good at generalizing from a few referent set members so as to recognize novel referents, and they are not satisfied with a word until they can do such a general recognition algorithm and usually come out with the same answers their neighbors do. But mechanical users of *gua\spi* cannot be expected to show such skill, and neither can beginning human users such as infants. They must build up a referent set for a word by exhaustively hearing referent set members. If an advanced human, or advanced software, can transcend the official definition of *gua\spi* words, that's fine — a common (but risky) strategy for humans will be to use their native language as a guide to *gua\spi* meanings.

However, *gua\spi* words are still defined officially in terms of referent sets simply because this definition is known to be tractable both for theory and for practical implementation. A *gua\spi* referent set is perfectly suited to be represented as a Prolog database, if truncated to a practical size.

Interpreting Language Behavior

When you speak an argument in a nonsentence you call the listener's attention to its referents. For example,

> **^:i |jiw ^sper -fe -jiol** *A crocodile!*

When you speak a sentence or a subordinate assertion you do the same thing: you call the listener's attention to the members of its referent set. (Thanks to John Parks-Clifford, editor of *The Loglanist*, for this insight [TL43].) Thus in:

> **^:i |qnu !qo -jan /tara /jun !kseo |zey !ju** *John, the rat is after your cheese!*

your knowledge of the referent set of "jun-*hunt*" includes a member which John will want to append to the ones he knows, before the cheese is stolen. This is the ultimate meaning of the *gua\spi* sentence.

Organization

Now that we have an unambiguous parse tree made up of phrases, what shall we do with it? Modern theories of parsing are very good at describing the transition from input tokens to a parse tree, but they leave the subsequent use of that structure pretty much to ad-hoc patchwork, and *gua\spi* is no better. However, the uses of the parse tree can be divided profitably into two classes, organization and semantics. Semantics in *gua\spi* consists mainly of computing and updating referent sets, whereas organization refers to a collection of preparatory transformations including assigning sub-phrases to cases, handling imbedded sentences, replacing pronouns by their bases, and transforming compound words into sub-phrases.

Which Words Go in Which Cases

The tones of grammar deliver to the organizational syntax level, for each phrase, an ordered list of attached sub-phrases, which are the arguments of the phrase predicate. For example in **!tara /crw !kseo**, "tara-*rat*" and "kseo-*cheese*" are attached to "crw-*eats*" as sub-phrases and therefore are its arguments. In the simplest and most common variation the arguments fill a sentence predicate's cases in order by number, much like English and Chinese, so "tara-*rat*" fills the first case of "crw-*eats*" and "kseo-*cheese*" fills the second. In arguments the first case is left unfilled. This organizational syntax can be so simple because the grammar delivers unambiguous lists of arguments, whereas in English or Latin a combined syntax has to deal with both getting the arguments on the right predicates and getting them into the right cases, and so is a lot more complicated.

The root phrase is assumed, in the absence of special cue words, to be a sentence; thus its first sub-phrase fills its first case. All sub-phrases are assumed to be arguments with empty first cases, except if they have tones or prefixed cue words that make them subordinate or infinitive clauses.

Should it be inconvenient to have cases filled in order, *gua\spi* has ways to change the order. First, certain prefixes signify that the relation word is *"converted"*: a certain case is exchanged with the first and so brought to the front. This is most useful for arguments. For example in **zu -crw** the first and second cases are exchanged, and the referent of such an argument would be something occupying the second case of *"eats"* before conversion: the meaning is *"food"*. The second case after conversion would then be the eater: **zu -crw !xo -tara** means *"rat food"*. The most common converted meanings have words of their own, such as "kqu-*food"*. Here is a florid example of conversion, in which one word serves for a sentence predicate and five different argument predicates:

0. *dou*

X1 throws X2 to X3 from X4 via X5 *1.*

dou *Pitcher, projector, launcher*

2. *zu -dou*

Missile (e.g. a brick or ball) *3.*

za -dou *Target*

4. *ze -dou*

Firing position, pitcher's mound *5.*

zi -dou *Route, trajectory, flight path*

1a. *zo -dou*

Thrower (suppresses any automatic conversion)

Definitions show the case numbers as X1, X2, etc. A caselink or a phrase-relative pronoun (described later) that pertains to a particular case finds that case wherever in the argument list it has been moved by conversion. Similarly, if there are several conversions on one predicate (not recommended) the one closest to the predicate has effect first, and the next one exchanges some case, wherever moved, with the new first sequential case.

Second, an argument can be directed to a specific case by a *"caselink"* prefix. For example, take

^:i !qo -jan /fer !se -dowu *John carries (something) from the house.*

qo -jan is *"John"*; **qo** marks a foreign name. **fer** = *"X1 carries X2 to X3 from X4 via X5"*. Its arguments are **qo -jan** in the first case, but **se** links the next argument, "dowu-*house*", to the fourth case: the start point. The caselink **se** attaches to and is one level down from the sentence predicate **fer**, hence would have falling tone. The argument predicate **dowu** attaches to **se** as a compound, and hence has high even tone. Sequential arguments jump over cases filled by caselinks.

English and many other natural languages use a *"subject-verb-object"* word order with the actor first, but in *gua\spi* the predicate can occur before, after or among the arguments. A sentence start word, or in sub-phrases some other prefix word at the same level as the predicate, will always occur before all of the arguments and will provide a jump point from which their grammatical levels can be established.

Since listeners like subject-verb-object order you should use it when possible, but listeners also like to hear complicated phrases at the end of a sentence, and you can achieve this goal by judiciously moving the predicate, by converting it, or by delaying a complicated argument to the end of the sentence and using an explicit caselink word. In English, converting the predicate produces the *"passive voice"*, which has a somewhat different meaning than the standard word order. No such passivity attaches to a converted *gua\spi* predicate. It is a fact, though, that listeners like the actor to be first when it can be expressed in one or two words, and do not like it to be omitted — common mistakes when people use the English passive voice.

It is permitted to say one or more arguments in isolation. This construction is called a *"nonsentence"*. It begins with the usual sentence start word **:i** and the arguments, as usual, are one level down, but there just isn't any sentence predicate.

Sentences as Arguments — Infinitives

A *gua\spi* sentence or argument expresses a relation between specific referents, and this specific referent set member is called an *"event"*. (Frequently the sentence represents several similar events.) It is common for several cases of the predicate to be vacant: in the previous example the thing carried, the destination and the route were not specified. Nonetheless there must have been a thing carried, a destination and a route, and the sentence asserts a relation between all five arguments. The next organizational elements we will look at are linking words that attach sentence predicates (with their arguments). The linked sentences represent lists of specific events with specific argument referents and with all cases filled even if not specified by words.

Returning to organization, the first sentence link word is **vo**, which acts to convert a sentence into a one-argument predicate, referred to as an *"infinitive"*, which means that the occupant of its first case is an instance of the sentence relation. Though **vo** can itself be a sentence predicate it is much more commonly used in arguments, like this:

> ^**:i !ji /vyu !vo -qia !ji** *I enjoy my bath*

vyu means *"X1 enjoys doing (*vo*) X2"*, where the second case is some kind of activity — a natural place to fill with an infinitive. The sentence linked by **vo** is **qia !ji** = *"I bathe"*, and an instance of that relation, an event, is the referent of the argument **vo -qia !ji** = *"my bath"*.

vyu includes the prefix **vo** on its second case by default, as do all words which commonly have infinitive arguments. Also, such words have various patterns, specified in the dictionary, in which main sentence arguments are replicated into infinitives. The most common is for the argument just before the infinitive to be replicated into the infinitive's first case, if the infinitive has no argument caselinked into the first case with **so**. Here **!ji** is replicated. So you could say

> ^**:i !ji /vyu !qia** *I enjoy bathing*

The extensive defaults on structure words, of which the default *vo* is one of the more common examples, increase the efficiency of *gua\spi* by letting the speaker not say most structure words.

Subordinate Clauses

A subordinate clause, indicated by the linking prefix **vu**, is a sentence within a sentence. Its most common use is to restrict a phrase (an argument or a sentence), so that a thing can be a referent of an argument only if it actually fits in the subordinate sentence, or the main sentence represents only events that fit in the subordinate sentence. Subordinate clauses are more common in *gua\spi* than in English, and also can be complicated, so several special rules are provided to make them simpler:

- Because subordinate clauses are so common the tone | is allocated specifically to them which automatically supplies the linking prefix **vu**. When this tone does not apply, of course, **vu** may be used explicitly.
- When the predicate of a subordinate clause has a case for an event, indicated by default **vo** or **bi**, the predicate is automatically converted to put the event first.
- The restricted phrase is automatically replicated in the first case of the clause which, if the previous rule applies, will be the event argument.

Here is a subordinate clause restricting an argument:

> **^:i !xi -ftu -plyw |xgi /fi -qke** *Green apples are sour*

> **^:i -qke !xi -ftu -plyw |xgi**

Not all apples (*ftu -plyw*) but only those which are green (*xgi*) are described as being sour (*qke*). The restricting sentence is *"X1 is green"*, and argument referents (apples) are automatically plugged into X1. The second version of the sentence is re-ordered to sound better; subordinate clauses usually do better near the end of the sentence. When in English we use adjectives and adverbs, in *gua\spi* we usually use subordinate clauses like this one.

Here is a subordinate clause restricting a sentence:

> **^:i !tara /cie -pne !kara ^vu -tum !vden !xgno** *The rat makes a hole in the box with its teeth (rat cut-penetrate box using teeth its)*

The restricting sentence is **!vo -X1 /zu -tum !vden !xdro** = *"[X1 is done] with its teeth as a tool"*, and the asserted relation **!tara /cie -pne !kara** = *"The rat penetrates the box"* is also required to satisfy the subordinate clause. The effect is as if an additional case were added to "cie-*cut*" for the cutting tool. Note that **tum** auto-converts so that **zu** is not needed in the subordinate clause.

The additional cases produced by subordinate clauses like this are called *"modal cases"*. They specify tenses, locations, listeners (vocative case), speakers in dialogue, repeated actions, and numerous miscellaneous cases as in the previous sentence. These cases are the *"context"* of a sentence. *Gua\spi* handles the context in a well-defined manner, whereas other languages handle context informally. As with numbered cases, something must fill every modal case in each event even if no words specify what that thing is. For example, many events are done *"by means of"* something, though rarely do we put words to the instrument. Many predicates in the language can give rise to modal cases. Therefore a predicate potentially can have a near-infinite number of cases.

Gua\spi has two other clause link words: **va** for subordinate assertions and **vi** for decorations that show the relation between sentences and the speaker's attitude about a sentence. Their syntax is the same as **vu**. For example (*go* being a mood marker for negation),

> **^:i |vi -csn ^tara |va -go -cul !zu -crw /fi -go -crw !kseo** *Strangely, the rat, which was not full of food, didn't eat the cheese*

> **^:i |vi -tan /pur -far !tara |zey -ji** *Damn, my rat ran away*

In "tan-*annoy*" of the second sentence, who is annoyed? "ji-*me*" is provided by default in the first case (before conversion) of any subordinate clause or top-level sentence whose first case ends up vacant, like this one. Thus top-level exclamations also become more natural:

> **^:i -fel** *Hooray! (I am happy)*

Quoted Text

A special argument form is quoted text. A quote is a prefix that transforms the following word or phrase to mean *"X1 is an instance of <something> as speech or writing"*. Here are the quote words, with the <something>s and with examples. The quoted phrases are underlined or in italics.

> **bu**

> **Word or words with compounding tone (high even, `-')**

> **^:i !xi -bu -ster /stu !jai |jir !xi -za -skul** *Shit is unsuitable to be said at school*

> **bi**

> **Phrase (sentence), with its arguments and clauses**

> **^:i !zglo /kam !bi !ji /daw -tao !term !ji** *He/she cried, I want (to be in contact with) my mother*

> **bo**

> **Phrase, but which is stated only approximately**

^:i !ken /jai !bo -juy *The boss (captain) says yes*

bn

The referent of the next argument, as text

^:i !bn -jw |zo -stul /fi -zu -srn !zglo *This letter (its content) is his response*

be

Words, not necessarily gua\spi, up to endmark ba

^:i !qo -kirka |ken /fi -jai !be C'est la vie ba *Captain Kirk said, C'est la vie*

br

Slash string (see text)

^:i !qo -:amlet /jai !br xa To be or not to be . . . xa *Hamlet said, To be or not to be . . .*

The last example, the *"slash string"* quote **br**, is the same as **ba** except that an arbitrary word (**xa** in the example) comes before and after the quoted text, in case **be** cannot be the endmark because it occurs in the text. **ba** and **br** are actually interpreted as part of the grammar, as very special cases, while the rest are recognized at the organizational level.

Story dialog is represented by quoted sub-phrases in English, but in *gua\spi* the dialog is at the main level and the speaker and listener are identified by a modal case with *"*jai-*say"*, *"*kam-*cry"* and related words. These words are defined as *"X1 says text (*bi*) X2 to X3"*, with **bi** as the default prefix. Since **bi** is an infinitive prefix, **jai** automatically converts in a modal phrase so the sentence is first. For example,

^:i |jai !qo -kirka |ken /qo -sulu |dri =cana /fi !jo /qma -sao !duwi |zu -tou *Captain Kirk said to Helmsman Sulu, Activate warp engines*

Later on, some unique features of *gua\spi* modal cases are illustrated in connection with story dialog.

Pronouns Represent Words, Not Things

The next organizational issue is the pronoun. In English, pronouns have referents just like any other argument. But *gua\spi* pronouns represent words, not the referent of words. In computer terms, they are like macro-instructions rather

than in-line functions. The represented words are called the *"antecedent"* of the pronoun, and the sentence is analysed as if each pronoun were taken out and replaced by its antecedent. The antecedents, not the pronouns, have referents. In this way the organizational syntax level can be kept free of meaning, and the semantic level has to deal with only one class of words, predicates.

For example, a document typically will have a signature line saying in effect *"this text is the output of Jim Carter"*. (Spoken discourse is completely analogous.) Then when there appears the pronoun **ji** (*"me"* in English) the effect is as if the words *"Jim Carter"* had been written in its place. That is, *"A rat ate my cheese"* and *"A rat ate Jim Carter's cheese"* mean exactly the same thing.

Phrases attached to a pronoun replace or supplement phrases in its antecedent. In general, if two arguments are given for the same case, numbered or modal, it is not an error but the first is ignored.

Phrase-Relative Pronouns

Gua\spi has phrase-relative pronouns, question pronouns, names and modal pronouns. Phrase-relative pronouns are for copying neighboring phrases — arguments or entire sentences. Each point of use may have a different neighbor, so the same word used in different places may have different antecedents, while to reach the same antecedent from different places may require different words. The pronouns whose antecedent is a whole phrase are automatically infinitives; an explicit **vo** is not needed. The most often used such pronouns are:

zdmo	First case of phrase where pronoun is
xgno	First case of current sentence
xdro	Referent of restricted argument
xdry	Phrase restricted by subordinate clause
zglo	First case of previous sentence
zgly	Entire previous sentence
vgry	Question sentence being answered
zgln	Previous discourse in general

There are pronouns for five cases and the whole phrase, for five phrase types — 30 in all. The complete list is under Vocabulary: Phrase-Relative Pronouns. Phrase-relative pronouns like *"it"* are common in English, but in *gua\spi* it is easier to redesignate an argument, so phrase-relative pronouns are less used in *gua\spi*.

Question Pronouns

For question pronouns the listener is supposed to say the antecedent; in other words, the speaker provides a sentence and the listener is to fill in the blanks. Here are the question pronouns:

	pu		What (argument)
		^:i !ju /crw !pu	What are you eating? (You eat what?)
		^:i !xo -ftu =plyw	An apple fruit (nonsentence)
	py		How (descriptive predicate)
		^:i -py !ftu =plyw	What kind of apple is it?
		^:i -xgi !ftu =plyw	A green apple (Green is apple)
	pa		How many (numeric predicate)
		^:i !ju /crw !ftu =plyw |zu -pa	You ate how many apples?
		^:i -co !ftu =plyw	One apple
	pr		Which one (identification predicate)
		^:i !ju /crw !ftu =plyw |pr	Which apple are you eating?
		^:i !ftu =plyw |zey !ju	Your apple (nonsentence)
	po		Is it (mood)
		^:i -po -qke !ftu =plyw	Is the apple sour?
		^:i ge — ^:i go	Yes — No

Names

In *gua\spi* a name is a pronoun. A name consists of a predicate prefixed by **qu**, or **qo** for foreign names, which disconnects the usual meaning of the predicate and substitutes the pronoun behavior. People are assigned permanent names at birth through a performative (ritual) statement like this:

> **^:i \|zo -tri ^qo -ben /xim !jw \|cil** *(Performative:) Ben is the name of this child*

From then on, **!jw \|cil** (*"this child"*, with context so the listeners remember which one) is the antecedent of the name **qo -ben**. Subsequently when Ben is introduced to people a speaker will make a similar statement, except lacking "tri-*ritual*" so it is constative (a statement of fact). Either statement lets the listener know what is the antecedent of the pronoun **qo -ben**. The antecedent then has a referent, which is the actual person being named.

The meaning (if any) of the predicate without **qu** is not changed. There is no implication that the antecedent may fit in the first case of the name, e.g. *"Mr. Coward"* may be quite courageous. A single referent can, and usually does, have several names. Semantics, though perhaps not law, allows you to put a new antecedent on a name at any time.

The six variables *da, de, di, do, du, dy* are names which you can assign to important concepts in nonfiction or characters in fiction. In mathematics it is common to use letter words as names for mathematical expressions, which in *gua\spi* have the form of arguments. When the name antecedent, or any pronoun antecedent, is assigned all pronouns therein are already replaced by their antecedents, so that if those contained pronouns are later reassigned the name antecedent is unchanged.

Personal Pronouns and Modal Defaults

Modal cases have a default value which is inserted in each sentence lacking explicit words for that case. Let us again use the speaker's modal case as an example. Most or all sentences in the document have the same speaker even though no speaker clause appears on them explicitly. There is a default value for the speaker, which is like a pronoun in that the antecedent words of the default are placed automatically into every sentence lacking an explicit speaker clause. The same default is the antecedent of the personal pronoun *"*ji-*me"*. The subordinate clause to identify the speaker initially is **|qi -jai !X**; the **qi** prefix indicates that the default modal case for the speaker should be changed to X. The default value is like a pronoun, in that the antecedent words of the default are placed into the sentences.

"Personal" pronouns, so called because *"me"* and *"you"* are among them, mostly represent certain important modal case defaults. There have already been several examples. The personal pronouns are:

ji *me*

The speaker *ju*

you *The listener*

je *we*

The speaker and the listener *ja*

we *Speaker and listener in imperative sentences*

jo *you*

The listener in imperative sentences *jn*

now *Current time default*

jr *here*

Current place default *jw*

this *Object being shown or pointed out*

jm *that*

Alternate object being shown *jy*

that which *The referent of the argument*

jy marks the open case of an argument, where the referent goes. It appears by default in the first case, but you may link something else there explicitly with **so** and put **jy** explicitly elsewhere. Conversion is neater.

The modal case default has a kind of stack, so that a previous default value can be saved, replaced, and later restored. Here is a list of the control prefixes for the modal cases and their stacks. Except **ql**, each must be used in front of a predicate, which identifies which modal case is being affected.

qe *The current default is saved and a new one put in effect*

qa *The current default is removed and the prior one restored*

qi *The current default is replaced by a new one*

qr *The current and prior defaults are exchanged*

ql *The speaker and listener cases are exchanged (for dialog)*

qy *X1 is the default modal argument for the predicate. The default is not changed; the sentence is simply true or false.*

Here is an example of stacked cases, in story dialog:

^:i |qe -jai !qo -kira /fi !py /zu-zni !cyr -far !ju *Said Kira, Why do you flee? (default saved, set)*

^:i -po -sfa -daw -can -siw -dan !ju *Don't you want to be rescued? (default inserted automatically)*

^:i |vi -pli ^vi -pql ^jo /kuo !ji *Please, at least talk to me! (default inserted automatically)*

^:i |qa -jai ^qo -kira /jun !suy *Kira pursued the swimmer. (prior default = narrator)*

Kira said the first three sentences, and the effect is as if the words **|jai !qo -kira** were added to the second and third, while the narrator said the last one, and **qa** restores him as speaker.

Tenses are also handled this way, as is any modal case. If you put **|qe -cnu !X** on the opening descriptive sentence (where X is an event identifying when the sentence happens) then it will be propagated to subsequent sentences automatically — unlike in English where a syntactically complicated and less precise tense has to be used on every sentence. John Parks-Clifford, then with the Loglan Institute, originally developed this concept of tense defaults [TL43].

In the third dialog line above, notice the use of *"jo-you"* to make the sentence imperative. The decoration *"|vi-pli-please"* shows the speaker's attitude but does not by itself make the sentence imperative.

Compound Predicates

A key organizational element of *gua\spi* is the compound word, a sequence of predicates and structure words. The first word's tone shows where the compound attaches. There may be several following words in the compound, each of which has high even tone - or low even tone =. A phrase predicate often begins with prefixes, such as the caselink **sa** or the conversion **zu**, but the most significant use of compounding is to join two or more predicate words.

The motivation to make compounds is twofold. First, you can use a single argument list to say what amounts to two sentences, which when compounded are much easier for the listener to interpret. Second, just as we use Latin prefixes in English to make many words from one, e.g. *"ob-ject"*, *"pro-ject"*, *"in-ject"*, *"ab-ject"*, most meanings in *gua\spi* are achieved by combining a much broader range of predicates. A beginner can learn the primitive words, about 1400, and then stick them together in self-created compounds which he can expect any listener to understand, while to achieve the same range of expression in natural languages the speaker and the listener must master a huge vocabulary in which most of the words are rarely used.

Though humans like to think of compound predicates as separate words analogous to the primitive words, compounds are actually defined through the transformations described below, so that each primitive word heads a separate phrase. For example in a transitive compound the compounded object is to be taken off and put in its proper case as a sub-phrase. Thus one can easily and reliably interpret a compound word that one has never heard before, as long as one knows all the primitive words.

Infinitive Argument

There are three main patterns to the compounds. First, if the main word has a case with a default linker of **vo** or **bi** — that is, a case for an infinitive — a word compounded with high even tone - is the predicate of that infinitive, and the main word case before the infinitive (before conversion) becomes the infinitive's first case. (Exceptions are noted in the dictionary.) Here are examples:

^:i !qo -kira /can -xna !fyni		Kira takes hold of the oar
	can	X1 changes so (*vo*) X2 becomes true
	xna	X1 holds X2 with (body part) X3
	!qo -kira /xna !fyni	Kira holds the oar (X2 of */can*)
^:i !du /juy -xna !do ^qnou !du		He offers it his hand
	juy	X1 allows X2 to do (*vo*) X3 (offers)
	xna	X1 (*!do*) holds X2 (*!qnou !du*) with (body part) X3
	!do /xna !qnou !du	It holds his hand (X3 of */juy*)
^:i !du /qma -jur !do		He turns it over
	qma	X1 makes X2 do (*vo*) X3
	jur	X1 (*!do*) turns by itself . . . (motion word)
	!do /jur	It turns (X3 of */qma*)
^:i -po -go -daw -siw -dan !ju		Do you want to be saved?
	daw	X1 desires to be/do (*vo*) X2
	siw	X1 stops being/doing (*vo*) X2
	dan	X1 is in danger of having (*vo*) X2 happen to it
	!ju /dan	You are in danger (X2 of */siw*)
	!ju /siw -dan	You are out of danger (X2 of */daw*)
	^:i -po -go !ju /daw (!vo !ju /siw (!vo !ju /dan (!vo !ju /X2)))	

The patterns with "can-*change*" and "qma-*make*" are as common in *gua\spi* as the pre-pro-in-ob-ab Latin prefix set is in English. In the second and third examples, the infinitive first case is a copy of the main sentence case before the infinitive: the second case. *Gua\spi* predicate words are usually not transitive, and the third example shows how transitive predicates are produced. (There is a separate word "fow-*force*".) As shown in the fourth example, multiple compounds are common. The expansion of this sentence is also shown with all defaults written out in full.

Shared Argument List

Second, the words may share an argument list. The effect is as if you had made two sentences with the arguments copied into each. This pattern is cued by **-** when the infinitive argument pattern does not apply, or by a conjunction **-fe** when it does. For example:

^:i !du /xna -zon !puou !do		He holds its head from behind
	!du /xna !puou + !du /zon !puou	He holds the head; he is behind the head
^:i !fnau /zu -vel !tfu -fen !do		The knife is in his harness
	!jy /tfu !do + !jy /fen !do . . .	His garment; which joins him to . . . (``*jy*'' = referent placeholder)
^:i !do /suy -pne -qmy !kqua		It swims down through the water
	!do /suy !kqua + !do /pne !kqua + !do /qmy !kqua	It swims to water; it penetrates that water; it is above that water.
^:i !du /trl -bil !do ^fcir		He drags it up to the air
	!du /trl !do ^fcir + !do /bil !fcir	He pulls it to air; it is below the air
^:i !do /suy -far !sa -du		It swims away from him
	!do /suy !sa -du + !do /far !X2	It swims from him; its destination is far

The first sentence shows this pattern plainly. In the second, the predicate of an argument is compound and the pronoun **jy** is used in the paired expansions to represent its open first case. It is very common for a motion word and a directional property to share arguments, as in the last three sentences. In the fourth one, with a transitive motion word the directional property relates the thing that moves, not the actor, to the destination. The polarity of the direction is often reverse of English: *"swims down"* means it is "qmy-*above*" the destination. A better translation is *"swims from above"*.

Most usually the first and second cases of all the predicates go together. For exceptions as in the fourth sentence above the dictionary shows which cases merge.

Object of Transitive Predicate

A third pattern is found in which a transitive main word is followed by its object as a compound. It is cued by the tone =. In this pattern the main word's second case (before conversion) receives the sub-word as an argument, except that if the main word is converted so the second case is unavailable, the first case (before conversion) gets the sub-word. For organizing the other arguments the effect is the same as if an explicit caselink **!su** had been used. For example:

	^:i -spo !bri =kqua |bir ^dri =fli		Maybe the pilot already drowned
		bri	X1 breathes X2
		kqua	X1 is a serving/portion of water
		bri =kqua	X1 drowns
		dri	X1 drives X2 to X3 . . . (transitive motion word)
		fli	X1 flies to X2 . . . (motion word)
		dri =fli	X1 drives flyers (airplanes) to X2 . . .

dri -fli would be the second type of compound, meaning that its referent both drives and flies, like a bird driving a car. This is not quite the right meaning.

Foreign Words and Metaphors The prefixes **fo**, **fn** and **qo** start a foreign word, and all subsequent compounded words are part of it. The word must start with a C and end with a V, and so you may have to modify the word either by removing or adding letters. Put a glottal stop before an initial vowel. Sounds not in *gua\spi* are mangled to fit. **qo** is for foreign names, **fo** marks a non-name predicate, and **fn** is for foreign metaphors. As a predicate the foreign word means *"X1 is (whatever)"* but listeners rarely know its meaning, so cases are never recognized on a foreign word. Normally you attach it as a metaphor to a *gua\spi* predicate, both to provide cases and to give the listener a cue, if not to the exact meaning, at least to the category of the meaning. For example,

> **^:i !qo -jan /dwu -fn -:au-stralo-pi-te-ku** *John is an australopithecine*

> **^:i !dlau -fn -borneo /juo !xr -bror -fn -:ma-ka-gani** *Mahogany trees live on the island of Borneo*

Here the foreign name *"John"* could be pronounced without change (note the phonetic spelling), but *"australopithecine"* needed work. While in English we use the adjective or set-membership form of the word, in *gua\spi* it works better to use the root form of the foreign word, if you know it: *"australopithecus"*. A glottal stop **:** was added at the beginning, *'th'* was changed to *'t'*, the final consonant *'s'* was removed (a V could have been added), and the word was stuck on "dwu-*animal"* to provide cases. This long word has five *gua\spi* syllables, treated in morphology as separate words, and they are stuck together by compounding tone **-**. (Normally the **-** tone symbols are not written inside a foreign word.)

Similarly on the other sentence, *gua\spi* predicates are put in front of the foreign words to help the listener interpret them. An educated human knows what a Borneo is, but a naive listener, particularly mechanical, needs the assistance of "dlau-*island"*.

Foreign words make the organization ambiguous: when a word has compounding tone is it compound or does it continue a foreign word? The Procrustean method is used to resolve the ambiguity: a foreign word eats all following first tone words.

While metaphors are most common with foreign words, they are also available with *gua\spi* predicates, being cued by the conjunction **fw**. They resemble shared-argument compounds but the meaning is not so precisely derived from the combined words. Only the cases of the main word appear; the sub-word's cases are ignored. (But metaphors are clearer if the first cases of all the words can merge.) Use regular compounds if at all possible, as in the last example, because they are unambiguous.

	bror -fn -:ma-ka-gani		Mahogany tree	
	dvu -fw -tal -xgi	Animal jumper green	Frog	
	kem -fw -kql -gnyr	Chemical liquid silver	Hydrargyrum	
	kem -fw -gnyr -kua	Chemical silver fast	Quicksilver	
	xgm -tsu -zu -jeu	Bright sudden shock (not metaphor)	Flash	

Special Cases

Numbers are a special case in organization. Compounded digits after the first build up a multi-digit cardinal number, with a possible sign, exponent and decimal point. Quasidigits, which appear first, mean *"approximately"*, *"at least"*, and so on. As a predicate a cardinal number means *"X1 is a set of N members, one of which is X2"*. An ordinal number, cued by the quasidigit **tr**, means *"X1 is N'th in list (*xy*) X2 starting at X3"*. For example:

> **^:i !tor =cenu /cni !ti -kl -co -cw -cu -ka -kn -ku** *The account balance is about minus 12.8 million dollars (about $–1.28E7)*

Conversion of compound words is a bit tricky. The conversion prefix carries the tone for joining to preceeding material. If the converted predicate has compounding (high even, -) tone, it alone is converted. But if it has sixth (low even, =) tone, it and all compound attached words are converted as a unit. (There is no provision for leftward grouping.)

That is how *gua\spi* is organized. Let us now turn to the semantics of arguments.

Argument Referent Sets

As stated earlier, a predicate word expresses a relation between the occupants of its cases, and is defined by a referent set consisting of lists of case occupants that are thus related.

To interpret an argument, you start with its predicate's referent set. You retain members consistent with any sub-phrases. From each member you extract the first case occupant, and out of these you make the *"full referent set"* of the argument. (The full referent set is empty in important special cases.) The *"referent subset"*, which is the set of actual referents of the argument, is a subset of the full set which depends on a prefix word called an *"article"*.

"Consistent with sub-phrases" means this: the n'th case occupant of each main predicate referent set member must be in the referent subset of the n'th case sub-argument. In addition, if the sub-argument has a prefix *"*ve*-each"*, then make equivalence classes of main referent set members (that survive all restrictions simultaneously) which differ only in the n'th case. If the set of n'th case occupants from an equivalence class is not equal to the sub-argument referent subset (i.e. it is not true that each sub-argument referent is in the equivalence class), reject the whole equivalence class. For subordinate clauses, the same procedure applies except the predicate is from the clause and the restricted phrase acts as one of its sub-arguments.

The Articles

Here are the articles. There are two articles for each meaning; the first unfolds the referent subset so each member is a referent, while the second specifies that the referent is the referent subset as a set. The careful distinction between sets and extensions of their members is characteristic of *gua\spi*.

xe, **xy**:
The most common article is **xe**, and it is assumed with most predicates when arguments lack an article. Its English translation is *"the"*. The referent subset is whichever members the speaker has in mind to talk about, but generally there are prior context cues to show which out of numerous possibilities are intended as the referents. In particular, if a set of referents has been designated before and if it is the only such set that is a subset of the full referent set of the argument, then those are the referents of the argument. For example,

> **^:i !fkar |xda ^vu -xge /fi -can -tai !qel =fkar** *The old, black car emerged from its garage.*

> **^:i !fkar /cyr -vle** *The car turned left.*

*"*fkar-*car"* appears three times; in each instances its article is *"*xe-*the"* but the article is unseen, being provided by default. The first instance of *"*fkar-*car"* designates one referent in detail. The other two instances are typical arguments with *"*xe-*the"*: since the prior referent fits this predicate (and in the second sentence *"its garage"* does not), the prior referent is being redesignated. Because *gua\spi* words are so short it is just as efficient to redesignate an argument like this as to use a phrase-relative pronoun, so pronouns are less commonly used in *gua\spi* than in English.

xa, **xu**:
The subset equals the full referent set. The referents are everything that fits the predicate. Mathematical expressions almost invariably have **xu** as the article, as in **xu -cu** = *"the equivalence class of all pairs"* = *"the number two"*.

xi, **xr**:
Only a few members are left out, which the speaker has in mind as being *"atypical"*. By judiciously using **xi** the speaker can prune out excessive special cases from his discussion. But unfortunately listeners have a lot of trouble to identify correctly which members are being left out. Beware.

xo, **xw**:
From the full referent set one or more members are selected, and it doesn't matter which ones. For example,

> **^:i |vi -pli ^jo ^sa -ji /gey !xo -kliw** *Please give me some nails*

All in the box are equivalent and it doesn't matter which you get. **xo** is often used for arguments in the *"serving or portion"* category, called *"partitive nouns"* in English.

xn:
No members of the full referent set are in the referent subset. This article is useful for negative statements like

> **^:i !tara /crw !xn -kseo** *The rat eats no cheese*

> **^:i !xa -kseo /go -zu -crw !tara** *Every cheese is not eaten by the rat (for every cheese, the rat doesn't eat it)*

Actually **xn** makes a statement about the excluded referent set members; were it regarded as a statement about all the members of an empty set, there could be no examples or counterexamples and the sentence would assert nothing. The second example is the contrapositive of the first, and shows what **xn** really means.

Each case of each predicate has a default prefix, used with any arguments that do not have explicit articles. Most of these defaults are **xe**; **xu** is used with math operators because of the meaning of numbers; **xy** is used for cases that need sets; **xe vo** is common for words that deal with events; and **xe bi** appears in a few cases that need texts. In the latter two cases appearance of any article cancels both **xe** and **vo** or **bi**. **xy** defaults have a special behavior: they change to **xe** if the argument is already a set.

When an argument predicate has a default article for its first case other than **xe**, and when it is in a case with a **xe** default, the argument predicate's article is used. Otherwise if the defaults are unequal and not **xe**, the situation is ambiguous and the speaker is required to say the article he wants.

Cartesian Expansion of Arguments

The reason arguments are in a phrase is to select particular events from the definition of the phrase predicate. When there are several arguments in a phrase their referent sets are multiplied in the sense of Descartes (Cartesian product) to give the events in the phrase referent set: each of the first argument referents is paired with each of the second argument referents, and so on for more arguments. For example,

> **^:i !tara |zu -cu /fi -crw !kseo |zu -cu** *Each of the two rats eats from each of the two cheeses*

> **^:i !kmau ^fe -kani /crw !tara ^fe -gara** *The cat and the cow eat the rat and the grass (but which eats which?)*

The second example could easily be interpreted the same as the first, but if one knows that particular events are not in the definition of the predicate — cows do not *"*crw-*eat"* rats — one will normally discount such terms in the Cartesian expansion without thought. The arguments then merely select pre-known events from the definition. If the speaker wishes to assert that a cow ate a rat, he should put the unusual event in a separate sentence and should emphasize it.

Often within a sentence — this applies to top-level sentences, not just phrases — the same argument appears in several cases, either as a pronoun or explicitly redesignated. In each member of the referent set of the sentence the occupants of the replicated case are equal — they are not selected independently from the referent sets of the replicated argument, as they would be if the arguments were different. Sub-phrases restrict main phrase events one by one, and the occupant of a replicated case in the sub-phrase is also equal to the one in the main phrase. For example:

> **^:i !kmau -bia |zu -ca /fi -jun !daue =xgno** *The three kittens chased their tails; each one chased its own tail*

> **^:i !cil |duo !ple !scer |zey =cil /fi -zu -pny |jro** *The children who are not sitting in their own chairs will be punished (child sit other than chair of child . . .)*

There is one exception: if an argument is prefixed by **vl**, its referent is chosen independently of any other copies of it. **vr** as a prefix means *"*inter alia*"* or *"each other"*. Its meaning is like **vl** except that in the sentence's referent set the prefixed argument's referent may not duplicate that of other copies of the same argument.

> **^:i !vdr =cor /dou -kyr !couo ^vr -zdmo** *The members of the soccer team kicked the ball to each other; none kicked it to himself*

If an argument prefixed by **vr** or **vl** is replicated by a pronoun the same referent is used both places, but if the pronoun has its own prefix that one supercedes any in the antecedent and the referent is chosen independently.

In addition, the sentence start words **:u** and **:o** mean that the sentence should be interpreted coordinately with the one before it. The referent sets of the two sentence predicates are multiplied in the sense of Descartes and the argument set members are fitted into their respective cases the same as in a single sentence. In particular, arguments replicated in both sentences have the replicated cases equal in each referent set member.

**^:i !ftu =plyw /xgi ^:o -bal !qke !zglo
^:i !ftu =plyw /xgi ^:o -bal !qke !ftu =plyw** *Since the apple is green, then likely it is sour; the same apple is sour*

The same statement is shown twice; the first one uses a pronoun while the second repeats the argument **ftu =plyw** explicitly.

For statements of logic one may produce a nonsentence containing the arguments to be discussed, called *"prenex arguments"*, and then state one or more sentences about them, starting each sentence with **^:u** and repeating the arguments or using pronouns. Since the Cartesian product is not commutative when negation is involved, or with mixed universal and existential quantification, the speaker may be forced to produce arguments in an inconvenient order, and prenex arguments may sound better. Here is an example of prenex arguments:

^:i !xa -tara ^jy |ga -xim =da ^:u !da /vam !tara *For all rats, there exists some X such that X loves the rat*

^:i !xa -tara /zu -vam !jy *All rats are loved by someone (same without prenex)* A conjunction is a pair or list of similar grammatical structures which act as if they were one unit. We have seen the conjunction word **fe** in compound words, but it also works with arguments to combine several into one, just as when one argument designates several referents. Here is a tricky example from before:

^:i !kmau ^fe -kani /crw !tara ^fe -gara *The cat and the cow eat the rat and the grass*

^:i !dwu /crw !kqu *The animals eat the foods*

The two argument pairs are Cartesian multiplied to select referent set members of *"*crw*-eat"* just as in the second example the two arguments designate multiple referents which are Cartesian multiplied. See also Vocabulary: Conjunctions for other kinds of conjunctions.

Miscellany about Arguments

Formally the articles are described as prefixes, but it makes sense semantically to regard them as predicates with two arguments; a full referent set comes in (as a set) to the second argument and the appropriate referent subset comes out in the first. With this definition, for example, **xa** (full referent set in extension) would mean exactly the same as *"*xor*-set member"*. It is easier that way to interpret compound articles like **xo -xi** = *"any typical X"*. However, articles have to be prefixes grammatically so the argument's main predicate can have its own arguments before it as well as after.

Most Indo-European languages distinguish between genders and numbers of arguments. Like Chinese and English, *gua\spi* has no gender, though you may use a subordinate clause like *"*|fmy*-female"*. Number comes from the referent sets, not the grammar. You may specify the exact number of referents with a numeric predicate, like this:

^:i |vi -pli ^jo ^sa -ji /gey !xo -beol |fmy ^vu -zu -cu *Please give me two nuts (female screws)*

Vocabulary

A great deal of the machinery of language, which in natural languages is shared between the grammar and the vocabulary, is handled in *gua\spi* purely by words. Here is an explanation of how to say a wide variety of basic language patterns. Frequently I have thought that some form or meaning required a new primitive word, or even a change in the grammar, but it has turned out that existing words were more than adequate if creatively used. Make this fact your prejudice in similar situations.

A major difference between *gua\spi* and Old *Loglan* (and, I fear, *Lojban*) is that *Loglan* relies heavily on metaphor and on the human ability to understand metaphors, whereas *gua\spi* compound words mean what they do because specific rules say how words combine. To say it differently, the meaning of a *Loglan* compound arises from wishful thinking, not from rules designed into the language. *Gua\spi* can represent metaphors — efficiently too. But in my *Loglan* writings I found that about 90% of all compound words could be interpreted according to rules I uncovered in the definitions, which Jim Brown was following unconsciously. I expect that even more compounds will turn out to be lawful in *gua\spi*, in which the definitions are tailored to make compounding easy, rather than taking the semantically slippery form of a true metaphor.

Functional Categories of Words

Gua\spi words have a relation between function and morphology. The CV pattern is for structure words, a few pronouns, and digits. CVV is for *"real"* relations, what in English are verbs. CCV is for abstract *"nouns"*, normally used to denote objects. CCVV is used for more concrete *"nouns"* such as species, chemical elements, or household artifacts. The purpose of these assignments is to cater to the known proclivities of natural language speakers, who like to segregate nouns from verbs, and to simplify the process of making words for meanings. Nothing in the language depends on these assignments, and the language never actually distinguishes between nouns and verbs. If you find that some word has a morphological form other than what you expect, this is simply an effort to keep related words together, and it will have no effect on the efficiency of the language.

Here is the distribution of *gua\spi* words by functional and morphological category. There are 11 V's, and 14 regular C's. **:** is allowed only in CV words such as **^:i**, and `#' does not count towards differences in words. There are 74 allowed CC digraphs out of 196 possible pairs; many CC's are too hard for people to recognize reliably.

Category	Form	Quantity	Total
Prefixes	CV	61	61
Pronouns	CV,CCVV	54	
Digits	CV	26	
Other structure words			80
Primitives	CVV	568	
Primitives	CCVV	532	
Letterals	CCVV	104	
Predicates	<TD> <TD> 1410		
Total words	<TD> <TD> 1551		

Form	Used	Available	Fraction Used
CV	98	165	59%
CCV	206	814	25%
CVV	568	1694	34%
CCVV	679	8954	8%

Word Creation

The words of natural languages appear to be arbitrary symbol strings of tremendous variety of sound. *Gua\spi* is similar in that its words were generated by a partially random process. To begin, the word lists of *Loglan* [L4] and *Lojban* [Lja] were merged and some additional words were added. For most words an English, Chinese and Latin translation was determined.

Then experimental phonetic data [NB2] was used to assess candidate words for the ease with which speakers could recognize them. For each *gua\spi* meaning, randomly generated word candidates were evaluated for recognizability, for distance from other *gua\spi* words, and also for similarity to their natural language equivalents. The final assignments were determined through a process of numerical annealing so as to maximize the summed quality scores.

Because the words are modestly similar to their natural language counterparts, learners are helped a bit in remembering them; and the sound patterns are anchored to forms known to please humans, since a prior attempt with purely random sounds was unacceptably ugly. The base natural languages included English because it is very widely spoken, Chinese (Mandarin) for the same reason, and Latin as a proxy for the other European languages, all of which have been influenced

heavily by Latin. (Latinoid English words were avoided.) Both *Loglan* and *Lojban* use many more natural languages as word creation fodder.

CV structure words were assigned by hand; related structure words, like articles, have the same C and varying V's. Structure words pertaining to numbered cases have the same V's as the corresponding digits, but contrasting consonants, making learning easier.

A question often asked is, why create new words? Why not use Chinese or English words? First, some attempt has been made to keep *gua\spi* culturally neutral, and if Chinese words were used it would intimidate English speakers and vice versa. More important, Chinese words are designed for use with Chinese. Many required meanings, like articles, simply do not exist in Chinese, and similarly in English. And those meanings that are present are only approximations of the *gua\spi* meanings; while users have to invest a lot of effort to learn the new words and their definitions, they will find it even harder to keep straight what a word of their native language *"really"* means in *gua\spi*. That is why the approach was rejected of simply stealing natural language vocabulary.

Given some set, a *"basis"* is a subset from which all its members can be derived, as with vectors. Each vector space has a specific dimension, or number of basis elements, but words are not so simple. *Loglan* has about 1000 primitive words and it was intended that virtually any meaning should be achievable as combinations of these words; that is, the primitive words form a basis of nearly all meanings. Experiment proved that this intention had been accomplished for the most part, but that as anticipated, some areas were incompletely or imprecisely covered. The *Lojban* project of LeChevalier [Lja] is a continuation of Brown's work on *Loglan* [L1], and he has added about 300 primitive words, mainly about human emotion and interaction. For *gua\spi* I took over LeChevalier's primitive word list, with his kind assistance. I rewrote all the definitions to match *gua\spi* usage. I also added and deleted a small number of words to deal with specific *gua\spi* issues, and I expanded the scientific vocabulary in mathematics, chemistry, zoology, botany and agriculture. As a result, *gua\spi* has about 1400 primitive words. Some people are interested to discover just how few basis words we can get by with. However, I have experience with the *Loglan* word list and I have confidence in its ability to handle the required meanings; and while I do not believe it is minimal, I think it is fairly close. Thus I chose to use existing word lists for *gua\spi* rather than to try for radical pruning or *de novo* creation.

Words and Grammar

Phrase-Relative Pronouns The CCVV pattern is used for organized groups of noun-type words such as the phrase-relative pronouns. These words have a CCV part and a final V, which matches the V in the caselink and the conversion for the same case, as well as the corresponding digit.

1st part	Phrase where antecedent of pronoun is located
zdm	Phrase that pronoun is in
xdr	Phrase restricted by phrase containing pronoun
vgr	Phrase to which replies are directed
xgn	Enclosing main-level sentence
zgl	Previous main-level sentence
Last letter	Which case is represented
o	1st case
u	2nd case
a	3rd case
e	4th case
i	5th case
y	The whole phrase

Here is a list of all the phrase-relative pronouns. There is also **zgln** meaning *"the previous discourse in general"* and **zglr** meaning *"the event just finished or still continuing"*.

Phrase	1	2	3	4	5th	Case (converted) of phrase . . .
zdmy	*zdmo*	*zdmu*	*zdma*	*zdme*	*zdmi*	That the pronoun is in
xdry	*xdro*	*xdru*	*xdra*	*xdre*	*xdri*	Being restricted
xgny	*xgno*	*xgnu*	*xgna*	*xgne*	*xgni*	Enclosing top level sentence
zgly	*zglo*	*zglu*	*zgla*	*zgle*	*zgli*	Previous top level sentence
vgry	*vgro*	*vgru*	*vgra*	*vgre*	*vgri*	Question to be answered

Letterals

Letterals are words representing letters, which are built up on a regular pattern like the pronouns are. A letteral means *"X1 is an instance of the letter (whatever)"*. **zu -fma !xo -<letteral>** is the right way to say that something has the shape of a letter. To spell a word, compound the letters start to end. The result means *"X1 is an instance of something spelled (whatever)"*.

Among letterals there are different forms for V's and C's, and upper and lower case in two alphabets are supported. The first table shows stem forms arranged phonetically, while the second shows what to substitute for the asterisks to signal alphabets and cases.

p *psl**

f *fsl**

b *bzl**

v *vzl**

m ***lm*

u ***lu*

t *tfl**

s *sfl**

d *dvl**

z *zvl**

n ***ln*

o ***lo*

c *cfl**

q *qfl**

j *jvl**

x *xvl**

l ***rl*

i ***li*

k *kfl**

g *gvl**

w ***lw*

y ***ly*

e ***le*

: *:zla*

*#vla*

r ***lr*

a ***la*

To select cases and alphabets, use these for * or **:

	*	**	Alphabet	Case	
	a	bx	Roman	Lower	
	e	dx	Roman	Upper	
	y	vx	Greek	Lower	
	o	zx	Greek	Upper	

The relation of Greek letters to *gua\spi* Roman letters is basically phonetic; the arbitrary assignments below are marked by *. The **:** can be used for the aspiration mark, an apostrophe in Greek or *'h'* in English transliteration, as in *"Hellas"*. In principle one can also use a form like **tler -fn -:alfa** (the letter alpha) to refer to Greek letters.

p *Pi*

f *Phi*

b *Beta*

v *Psi**

m *Mu*

u *Upsilon*

t *Tau*

s *Sigma*

d *Delta*

z *Zeta*

n *Nu*

o *Omicron*

c *Chi**

q *Theta**

j —

x *Xi**

l *Lambda*

i *Iota*

k *Kappa*

g *Gamma*

w *Omega**

y *Eta*

e *Epsilon*

: '

—

r *Rho*

a *Alpha*

In English we use acronyms freely, but in *gua\spi* the letterals for the acronym are as long or longer than the words themselves. It is better to make an ordinary compound word for the concept.

Foreign Words

It is the policy in *gua\spi* to use foreign words as-is (except for necessary mangling to make them fit the CV pattern), to represent the names of foreign people, places, flora, fauna, units of measure, foods, clothes, and so on. Certain of these word categories have a few members assigned CCV words. These are in *gua\spi* because they were in *Loglan*, and they were in *Loglan* because they occurred with high frequency in European literature. In *Loglan* it has proven impractical to manufacture in-language primitive words for every possible primitive, not for lack of word space but rather because those working on the language have other issues to attend to than making a continuous stream of predicates. For example, such an obvious animal as *"bear"* didn't make it into *Loglan* and probably never will. *Gua\spi* will have primitive words for more fauna and flora, but as in *Loglan*, the majority of species will never have *gua\spi* primitive words, and neither will most ethnic foods, foreign countries, provincial units and so on.

When foreign words fit neatly into the language, speakers are tempted to over-use them. Agglutinative languages like *gua\spi* and *Loglan* have great power to produce compound words with quite precise meanings, and speakers should try

very hard to learn to produce such words. It is a fact that despite *Loglan*'s limited set of nouns and, at that time, its lack of foreign words, I was able to write about 20,000 words of text while inventing only four unapproved primitive words (and a fifth was made for me after the fact): bear, torus, tape, noodle and oar. Speakers of *gua\spi* should try to emulate this performance and to use foreign words only for truly local concepts, such as the *"|mu"* (a Chinese area unit for land) or *"adobo"* (a Filipino chicken stew).

See the discussion under Pronouns: Foreign Words about attaching foreign words to *gua\spi* predicates to provide cases.

Retroactive Downjumps

Sentences are usually connected by a retroactive downjump **^:o**. With this operator the preceeding sentence is taken out of the discourse and is inserted in the first case of the word following **^:o** (which will need a default **vo**). Thus the following two sentences are equivalent:

> **^:i -dae !kara ^:o -bal !crw |jro ^tara ^kseo** *If the box is open then maybe the rat will eat the cheese (With retroactive downjump)*

> **^:i !vo -ge -dae !kara !fu -bal !crw |jro ^tara ^kseo** *(With explicit infinitive)*

The form with the explicit infinitive is more natural in *gua\spi*. When retroactive downjumps are allowed you have to finish an entire sentence structure and hear the next word, possibly a downjump, before you can place the structure in the parse tree. However, all natural languages allow afterthought sentence connectives as in the above examples, and speakers rarely use forethought forms where the beginning of the antecedent sentence is marked. I wonder if users might resist an absolute requirement to put **!vo -ge** at the beginning of the antecedent sentence.

Speakers, please try to minimize the use of retroactive downjumps. We shall see if it is feasible to outlaw them completely.

Since the antecedent sentence starts out at the top level, it is asserted by default and remains thus after the downjump. The consequent sentence is an ordinary infinitive, which is not asserted. If you wish to assert it, or the antecedent when not using the retroactive downjump, put in a **ge** prefix as shown for the explicit infinitive antecedent.

A sentence start word **^:u** connects sentences with coordinated arguments, and such a group acts as a unit for retroactive downjumps. **^:e** also makes such units, but the arguments are not coordinated. More complicated groupings are best handled by making explicit infinitive arguments out of the component sentences and using **ge** to assert them, or by assigning names to the sentences and subsequently asserting a causal relation among the antecedents of the names.

The retroactive downjump **fy** relocates the sub-phrase before it at the same level, to become the first case of the following predicate at the same level. It roughly translates the infix *"and"* that connects a list of arguments in English. For example,

> **^:i !tara /crw !kseo ^fy -tla !qkao |spl /ftu =plyw** *The rat eats the cheese and the cookie and the apple (an unordered set)*

> **^:i !ji /vlw !qo -:nuarko ^fy -stl !qo -trentn ^qo -prinstn** *I travelled to Newark and Trenton and Princeton (an ordered list)*

This is one way to do multiple arguments (see also the section just below), and is just about the only use for **fy** — but difficult for hard-core English speakers to give up. Needless to say, the retroactive downjump is not mandatory. *"*tla-*set"*

is used when the arguments form an unordered collection, while "stl-*list*" is for arguments that have a sequence. Both words take as many arguments as you wish.

When a sentence is complicated, one can use "fl-*begin*" and "fr-*end*" to mark the beginning and end of a grammatical unit. A matching pair of **fl** and **fr** are supposed to be at the same grammatical level; if they are not, someone has made an error. They have no meaning beyond this checking function.

Error Correction

Another organizational transformation related to the retroactive downjump is error correction with **fa**. When prefixed to a phrase, it causes the previous phrase at the same level to vanish and to be replaced by the one that follows. **/fi -fa** is a quick way to delete the current sentence. For example,

> **^:i !ji /vyu !gunu !ju /fa !diu -sui |zu -ken !ju** *I like your ass, I mean, your class*

> **^:i -po !ju /crw !ftu =plyw |zey !ji /fi -fa** *Did you eat my apple . . . Forget it.*

Subphrase Forms

Conjunctions, Mixtures and Masses The conjunction **fe** has several functions depending on what it joins. First, when it joins phrase predicates it makes a collection of phrases with the same arguments; usually it is unseen in this function since the first tone - is usually sufficient to cue a parallel phrase compound.

> **^:i !jo /cyr -tai !sa-kqua** *Everyone get out of the water*

> **^:i !qu -jaiw =tiri /cur -cie !qka -pne !qno =cail** *Tigereye cuts a hole through the piece of steel*

Despite our monkeylike eagerness to treat these words as units, we should remember that in *gua\spi* there are phrases being conjoined. Thus everyone is told to go from the water *and* to be out of the water; and the result of cutting is both a hole *and* goes through the steel; the referent of this argument satisfies two predicates.

fe also can join arguments. The first of the following examples shows a conjunction with **fe** in which successive arguments are joined into one. (For details, see Semantics: Cartesian Expansion.) The second example shows the same sentence rendered with "tla-*set*" in which the set members are listed. **fe** produces referents only in extension whereas **tla** produces a set in suitable context. There is a similar word "stl-*list*" when the arguments must have a specific order. For both words you may give as many arguments as you need.

> **^:i !kmau ^fe -kani /crw !tara ^fe -gara** *The cat and the cow eat the rat and the grass*

^:i !tla !kmau ^kani /fi -crw !tla !tara ^gara *(The same with sets)*

Here is a particularly troublesome example of *"or"* in a set of arguments:

^:i |vi -pli ^jo /sle !xo -zu -co ^xo -psuw =fiei ^fe -xo -psuw =cawi ^fe -xo -qkmy *Coffee, tea or milk? (Please choose one from the list)*

Mixtures are expressed in several ways.

^:i !faia !do /fi -gai !xy -fpyl |sew !fkl ^qki *Its face was covered by blue-gray fur — in the overlap (sew) between blue and gray*

^:i !flyl /zar !dvr ^vba ^fkl *The flag is red, white and blue — some parts in each color (zar)*

^:i !jw /dmy !qkmy ^qtaw |zu -dma !xo -bimi *This is a mixture (dmy) of milk and honey (sugar from bees)*

Loglan has a concept of a *"mass individual"*. According to Brown [L1] it is more characteristic of non-Western cultures. Here is my best explanation of it. Take the full referent set of an argument, and personify it so that, potentially at least, it is the same kind of thing as its members. For example, all sharks can be considered to be instances or manifestations of an archetypical shark god. This composite object is the mass individual. In *Loglan*, arguments in the *"serving or portion"* category, like "cutri-*water*", generally are used as mass individuals.

Concepts and features which in other languages seem unitary are revealed in *gua\spi* to be various. The mixtures above are one example, and mass individuals are another. Here are some examples related to mass individuals. Note that **xo** is the default first case article for servings and portions and so in argument sites with the normal default you need not say **xo**.

!xo -kqua *A portion of water (any one)*

!xe -kqua *The portion of water*

!xe -psu -kqua *The water molecule*

!xy -fpyl *The hair (set)*

!xy -sur -fpyl *The fur (emphasizing repetition)*

Named Arguments

^:i !qo -:mobi -dyk /kqnu |bir ^sa -qo -german -:melvyl *Moby Dick was written by Herman Melville*

^:i !qu -jaiw =tiri /fom !qu -kmaw =za -tye =tlme |va -zo -kqnu ! sa -ji *Tigereye is a character in The Welding Shop, which I wrote*

^:i !qu -gua |xim !cfle /vu -zu -ziu !sa -kmi =xi -kmum /fi -pror |bir ^sa -xu -tla !qo -krnygan ^qo -:ryci *The C Programming Language</u> was written by Kernighan and Ritchie*

^:i !qu -vo -tri -qtu =jy /zu -xim !xgnl |qe -zo -zymu !sa -qo -gabriel -fora *Requiem by Gabriel Faur* *(title on the score)*

"Moby Dick" is a bit ambiguous; it names both the book and the whale in it. The fault lies with the author for using one name for two referents. In any case, it is obvious what a foreign name means when referring to an object or a person. In the second example we have two *gua\spi* predicate names, conveniently all compound words so that they feel the same as the foreign names. But in the third example the predicate name is quite intricate, extends over three grammatical levels, and includes an imbedded name, *"C"*, represented by a letteral. Nonetheless the principle is the same; **qu** converts the following phrase into a name.

The fourth example is perhaps the most difficult: the declaration of a title. The title is not part of the discourse but rather tells about it, hence it takes the form of a decoration with **vi** in a nonsentence. With it the by-line appears in the usual form for setting a modal case default.

Modal Cases

Here are examples of the most important modal cases. However, virtually any word with two or more cases can be construed as a modal operator. Be alert for creative opportunities for expression.

cnu	Present tense	
	^:i !ji /za -ven !su -fkar ^vu -cnu !jrn !ji ^pra !fkar	I will buy the car *when* I have earned its price
bir	Past tense	
	^:i !ji /crw !kseo ^vu -bir !jun -vnl !tara	I ate the cheese *before* the rat came hunting
jro	Future tense	
	^:i !ji /qma -klo !kara ^vu -jro !crw !tara ^kseo |zu -vel ! kara	I closed the box *after* the rat ate the cheese in it
jir	The location of an event	
	^:i !ji /fom !zu -plu |jir !pil -fn -:olimpu	I performed in the Olympic games, the games *at* Mount Olympus
zey	Genitive or possessive case: a relation of pertinence	
	^:i !trer =ji /dri !fkar |zey !ji	My brother is driving *my* car
jai	The speaker	
	^:i |jai !qo -kira /ju /dwu -csn -zu -jeu	*Said* Kira, ``You're a monster''
koy	The thinker, analogous to the speaker	
	^:i |koy !qo -kira /dri -fli /bni -siw -dan |qma	*Thought* Kira, ``The pilot needs to be rescued''
qnu	Vocative case: the listener paying attention	
	^:i |qnu !qo -jan /tara /jun !kseo !zey !ju	*John,* the rat is after your cheese
ciw	The holder of a subjective opinion	
	^:i !xi -tara /fpl |ciw !ji	Rats are beautiful, I *feel*
gae	The experiencer of an emotion (most have a case for this)	
	^:i !ju /csn |gae !ji	You're wierd, I *feel*
sen	The experiencer of an objective sensation	
	^:i !ftu -plyw /qke |sen !ji	The apple tastes [is] sour *to me*
brm	What something is part of (many parts have a case for this)	
	-xo -bryr |brm !dowu !ji	A brick *of* my house
tum	The instrument for doing an event	
	^:i !qo -kira /qma -pai !cana ^vu -tum !tuen	Kira bailed the boat [made it drain] *with* the bucket
zia	The way or manner of an event	
	^:i !xau =spa |cana /fi -fli |zia !fli !xi -bryr	The space shuttle (roundtrip spaceship) flies *like* a brick (flies)

There are several moderately complex cases, such as "tue-*culture*", and "fta-*standard for judgement*", which are relevant to only four or five words each. These are served by modal cases as a matter of policy, rather than having numbered cases on each word.

Tenses and Aspects

In English and all Indo-European languages, every sentence has a tense, that is, the syntax indicates (with great complexity) when the sentence occurred. Stories, for example, frequently have every narrative sentence in the past tense. *Gua\spi* uses instead the modal case default for tense, and explicit tense modal cases appear only for sentences off the default. The sentence start word **^:a** links sentences that occur in sequence. Here are some example sentences with tenses.

> **^:i !tara /crw !kseo ^vu -cnu !vn -juw -qana** *The rat ate the cheese at midnight*

> **^:i !tara /crw !qkao -spl ^vu -jro !xdry !su -kseo** *The rat ate the cookie after [it ate] the cheese*

> **^:a !kmau /crw !tara** *Then the cat ate the rat*

> **^:i |qe -bir |fto ^jn /vu -qe -jro |pql ^cyr -xyn !qo -kaesar ^qo -gal** *A long time ago, Caesar had just entered Gaul (a nonsentence, it asserts nothing but does set the tense default)*

In Russian, every verb is formally assigned an *"aspect"*: *"perfective"*, meaning that the sentence's event is considered as a unit, including its completion, and *"imperfective"*, meaning that the predicated relation is continuous. English has these aspects too, though each verb can have either aspect depending on a moderately complicated syntactic cue (*"ing"* for imperfective and various others for perfective). Speakers also like to distinguish a *"completed"* versus *"aborted"* aspect, whether an event reached its usual conclusion. In *gua\spi* the unmodified predicates are perfective or imperfective according to their meanings, but subordinate clauses or compounding words can express whatever aspects are necessary. Here are a few aspects in *gua\spi*:

> **^:i !xo -kqua /tiu -flu !se -bil !bryo** *Water continuously flows under the bridge (imperfective)*

> **^:i !tara /crw |scu ^kseo** *The rat completely eats the cheese (perfective)*

> **^:i !kmau /qem -jun -xna !tara** *The cat tries to catch the rat (might or might not succeed)*

> **^:i !kmau /qai -jun -xna !tara** *The cat fails to catch the rat (though it tries)*

Decorations

A decoration is a short subordinate clause. Sometimes it expresses the speaker's attitude about the sentence or the relation between sentences, in which case its prefix is **vi**, or it can be a subordinate assertion with **va**, or an actual part of the main assertion, with **vu** or | tone.

bwy	On the other hand; Contrast	
	^:i !ji /pql -gal ^:i |vi -zo -bwy ^ji /bzu	I am short; *on the other hand*, I am wide.
smy	Similarly, Likewise	
	^:i !ji /jie -crw ^:e |vi -zo -smy ^ji /jie -byw	I am hungry, and likewise I am thirsty
csn	Strangely	
	^:i |vi -csn ^tara /go -crw !kseo	*Strangely*, the rat didn't eat the cheese
spo	Maybe	
	^:i |spo ^kqu /zu -vel !kara -zgy |zey =ji	*Maybe* food is in my refrigerator
zba	Likely	
	^:i |zba /fpu !kqu |zu -vel !kara -zgy |zey =ji	*Probably* the food in my refrigerator is spoiled
tfn	Importantly	
	^:i |vi -zo -tfn ^jo /ser -sui	*Importantly*, you must study
tsu	Suddenly	
	^:i |qe -jai !do ^ve -tsu ^ve -fto -faw . . .	He says *suddenly* and very emphatically . . .
jiw	Surprise	
	^:i |va -jiw ^cana -fer /vnl	*Hey*, the barge is coming
pli	Please	
	^:i |vi -pli ^jo /pin -dwo	Please be patient
gny	Kindly	
	^:i |vi -gny ^jo /gey !xo -kqua ^ji	*Kindly* give me some water
faw	Exclamation Point	
	^:i |faw ^jw /pei -bil !ji ^kqua	He dunked me!
din	Do Back	
	^:i |ql ^jo /pei -bil !jw |din /kqua	So dunk him *back*
gza	Start of Paragraph	
	^:i |vi -gza ^qu -jaiw =tiri /qou !dvu |jir =zna	¶ Tigereye watches the waves in the canal
qti	Thesis (Topic Sentence)	
	^:i |vi -qti ^gua =spi /pwo !xa jy |zu -pwo !gua =fn -:ewlan	*Gua\spi* can do everything English can do
sku	Thus, Conclusion	
	^:i |vi -sku ^zge |xne !gua =vdm ^gua =cin /fi -klo |va -ge -qma !gua !spi	Thus the gap between human and machine languages is closed by *gua\spi*

Mathematical Expressions

Even uneducated speakers quantify phrases, that is, they say how many or how big some phrase is. It turns out that to support just simple dimensioned quantities the language has to include a complete facility for mathematical expressions.

Numbers, Expressions and Functions

Cardinal numbers (here exemplified by **cu** - set of two) are defined as *"X1 is a set containing so many members X2"*. The converted predicate means *"X2 is a member of a set X1 of so many members"*. Quantifiers are subordinate clauses on an argument, e.g.

> **^:i !cil |zu -cu ^ji /fi -vyl** *My twins are male*

How do you say *"the number two"*? Any set with two members can be put in 1-1 correspondence with any other such set, but not with a set with different count; this forms an equivalence relation that segregates sets by count. Among the ways to define *"the number two"* the one that fits best in *gua\spi* is **xu -cu**, designating this equivalence class. All kinds of mathematical objects, such as rational, real, complex and dimensioned numbers, can be produced by various extension maneuvers from these equivalence classes, and can be named in *gua\spi* by **xu -N**.

> **!xu -cu -cw -ci** *The number 2.5 (the class of all sets of count 2.5)*

Mathematical functions are defined with such classes as formal parameters, and hence have **xu** on parameter cases by default — **xu** means the entire referent set of an argument, as a set (or class). The first case of a function is its value, and the function is defined as *"X1 is in the equivalence class that comes from doing (function) on (*xu*) X2"*, possibly with several parameters. Thus a function can be used to predicate that something has a particular count or measure. **xu** recovers the equivalence class. The abbreviation *"IEC"*, meaning *"in equivalence class"*, is used thus: *"X1 IEC the result of (whatever)"*. For example,

> **^:i !xa -ca /plw !co ^cu** *3 is the sum of 1 and 2 (all triplets IEC 1+2)*

This syntax for mathematical expressions is neat, compact and unambiguous. No special syntax needs to be added to *gua\spi* beyond that already in use for ordinary arguments and sentences.

Functions always deliver their value in the first case and take arguments in the second and following cases. For the range and domain of a function F, use **xu -F** and **xu -zu -F** respectively.

Ordinal Numbers and List Ends

An ordinal number, cued by the quasidigit **tr**, means *"X1 is N'th in list (*xy*) X2 starting at X3"*. For example:

> **^:i -brn !junu !qnou =ji |tio /ve -tr -ca** *Broken is the third claw of my right paw (hand)*

List ends and segments are built with *"*bny*-begin"* and *"*fne*-end"* restricted by a numeric predicate. Note the definition, *"X1 is the next or previous member of (xy) X2 after X3"*; restrict with a numeric predicate to change to the N'th next or

previous member. Without X3 the list ends are produced, but don't be confused by the polarity: "bny-*next also means ``beginning"* or *"least"* when the list is ordered by size or degree; "fne-*previous"* means *"end"* or *"most"*. It is clearer to use an ordinal number when you can. For example,

^:i |vi -pli ^jo /can -fne !psa -gvu *Please go to the end of the line*

^:i !ji /bny |cu ^sty -kqa !diu -sui *I am the second smallest in the class*

^:i !ji /tr -cu !sty -kqa !diu -sui *Same thing, better*

^:i !bny |te -ca ^sty -bir !kuo |tum =teon ^ji /fi -za -gey |jro ^su -xo -spia *The first three people (in order by time) to phone me will be given tickets*

^:i !tr -te -ca !sty -bir !kuo |tum =teon ^ji /fi -za -gey |jro ^su -xo -spia *Same thing, better*

Lists are ordered with smaller or negative numbers first, so the *"smallest"* is **bny -sty -kqa** whereas the largest would be **fne -sty -kqa** or, sorting the list in reverse order, **/bny !sty -spl**. See also the discussion of "sym-*chief"* under Comparative and Superlative for a better way to do *"second smallest"* and the like.

Vectors, Dates and Times

You express a vector as a "stl-*list"* of expressions. Units of measure applied to a vector multiply each component individually. A matrix (by components) is a list of vectors, and so on. A date or time is also a list of expressions.

^:i !vnyn /zm -cmu !dman ^dmem !stl !ci ^ca -cy *The wind is (5, 30) meters per second (per second meter 5, 30)*

^:i !qo -kauai:i /jir !vdei !stl !co -ke ^kl -co -ci -ko *Hawaii is at (19, –156) degrees*

^:i -tem =jani !su -jn ^stl !co -ke -ka -ke ^cu ^co -ke *The date today is 2/19/1989 (order: year, month, day)*

^:i !qo -kamleto /zu -fom |tem =qrau !stl !cu -cy ^ca -cy *Hamlet will be performed at 20:30 hours*

The date is defined as *"X1 is the date of event (vo) X2+ starting with unit (xu-jani) X3* in calendar X4"* in which auto-conversion lets it restrict a sentence directly, while the unit can still be compounded. The first vector component has that unit, and subsequent components are multiplied by sub-units in the order years, months, days, hours, minutes, seconds. The default unit is "jani-*years"*.

Units of Measure

Units of measure are defined to multiply a number or other expression by the unit. The resulting equivalence class is considered to contain *gua\spi* events whose degree or measure are that big; hence the unit expression takes the form of a subordinate clause, and the main sentence predicate tells what dimension is being measured. For example,

> **^:i !ji /vga |kyam !ku -cy** *I weigh 70 kilos (I heavy kilo 7 0)*

> **^:i !tor =cenu /cni !ti -kl -co -cw -cu -ka -kn -ku** *The account balance is about minus 12.8 million dollars ($ about −1.28E7)*

"Scientific notation" is used in *gua\spi* instead of the thousands and millions typical of English and in place of the metric prefixes; it is more compact and much easier to specify syntactically.

This definition of a unit is reasonable mathematically since a physical unit of measure can be interpreted as a basis member of a 1-dimensional vector space of things having that dimension. For example, consider mass. Take the set of all things with mass, and take equivalence classes of things with equal mass. Those equivalence classes occupy, and can be extended to create, a 1-D vector space. Any single member is a basis, and a unit is a member selected by convention, e.g. *the* standard kilogram. Now for the word, its referent could be the unit, but you have to multiply it by the number (e.g. 2.5 times grams), which makes expressions too wordy. So the unit word is defined as a math function that multiplies by the unit.

In units of measure, the first argument occupants are not things but properties, e.g. masses of things, which are events, e.g. *"something is massive"*. The need for a predicate to go with the thing being measured is easiest to see in 3-D, e.g. the argument could be high, wide or deep but all are measured by the single dimension of meters. Then the unit becomes a modal case of the predicate. These examples show how to use MKS and provincial units:

> **^:i !ji /gal |dmem !co-cw-ku-ce** *I am 1.74 meters high*

> **^:i !ji /gal |xnu -fn -:inca !ko-ke** *I am 69 inches high*

In particular, no quantifiable relation (e.g. *"heavy"* or *"exceeds in dimension* vo *X3"*) has an explicit case for how much it is, relying instead on the modal case of units. There is one exception: "kun-*quantity"* is like a unit in providing a modal case for quantity, but provides an identity transformation, so that a question word can be dropped into the multiplicand argument without forcing a specific unit.

To talk about the unit rather than to use it, use **xu vo <unit>**, as in *"the pound is a provincial unit"*. **xe vo <unit>** will deliver the standard unit, if there is one, given suitable context cues.

Compound units, like ohms, require a product or quotient of several units. One may use the personal name units (ohm, volt, pascal, celsius) in the same manner as provincial units.

Quantification and Negation

Some Important Quantifiers

xa -tara	*All* rats (anywhere, any time)
xa -xe -tara	*All* the rats (in an in-mind set)
xa -tara |xyn !dowu	*All* the rats in the house
xi -tara |xyn !dowu	*Most* of the rats in the house
tara |zu -vdu	*Many* rats
tara |zu -pqu	*Few* rats
tara |gou -sun	*Enough* rats
tara |gr -gou -sun	*Too few* rats
tara |gou -pqu	*Few enough* rats
tara |gr -gou -pqu	*Too many* rats (insufficiently few)
tara |zu -ti-ta-cu-cy-cy	*Almost* a hundred rats
jmo -vjr	*Almost* vertical

Words for Something

^:i !ji /crw |bir *I already ate (something implied)*

^:i !ji /daw -crw !jy *I want to eat something*

^:i !ji /crw !xo -kseo *I am eating some cheese*

^:i !jw /vdr !xy -jy |kfa /vu -sny *Logically, he must have some family (a set)*

^:i !xi -jy ^:u -xun !vo !zglo /gr -zu -gul ^vo !zglo /qma -tfa *Most things are illegal or fattening*

^:i !xa -jy |vdr !xo -sto -fw -kaia . . . *For anything in a compact set . . .*

Nine Varieties of Negation

^:i -sfa !kio !ji ^tara |zey !ju

It is false that I have your rat. This is the prototype of negation, and it is the policy in *gua\spi* to use predicates when possible rather than prefixes or other structure words. However, the negated sentence is an extra level down, a problem for speakers.

^:i -go !ji /kio !tara |zey !ju

I *don't* have your rat. **go** is a mood prefix which means that the asserted sentence is counter to fact. It is simpler and more familiar to natural language speakers than *"*sfa-*false"* is, and it works in subordinate clauses where **sfa** doesn't.

^:i !ji /kio !tara |go -zey !ju

I have a rat which *isn't* yours. **go** can equally be used in subordinate clauses, or even in argument predicates.

^:i !ji /kio !xn -kseo

I have *no* cheese. **xn** means that of the members of the full referent set of the argument, none fit in the predicated relation. Unlike the rest of the articles, this is actually a statement about the excluded members, and means the same as **^:i !ji /go -kio !xa -kseo** — freely translated, *"for all pieces of cheese, I don't have it"*. (See De Morgan's rules below.)

^:i !ji /kio !kseo |zu -cy

I have *zero* pieces of cheese. This is the most natural form of argument negation in *Loglan*, but *gua\spi* looks strictly at referent sets, and if you say you have all the members of the null set, it isn't a cheesy null set — there is only one null set. The statement is a tautology, and says nothing about cheese. Many logical fallacies, such as St. Anselm's ontological proof of the existence of God, are like this example in that they prove a statement about the members of a set which may not have any members. In *gua\spi* use **xn** as above.

^:i !ji /kio !ple !tara

I have something which *isn't* the rat. The full referent set of **ple !xe -tara** (and therefore its referent subset) is in the complement of the referent subset of **xe -tara**.

^:i !jw |kseo /fi -stu -zao

This cheese is *bad* in flavor. In George Orwell's 1984, the language *"newspeak"* was designed to destroy the ability of people to think, and one of its design features was that negative words were eliminated; *"bad"* became *"ungood"*. *Gua\spi* (imitating *Loglan*) offers specific negated words for major predicates when the negations are used frequently. Nonetheless, most negations will have to be done with compound words as in the next examples. Be alert for creative expression possibilities such as **^:i !jw |kseo !fu -zu -dyi** — *"this cheese is* disgusting*"*.

^:i !jw |kseo !fu /gl -zao

This cheese is flavor*less*. Many dimensions are quantifiable (more or less) but unsigned, so their degree ranges from zero to larger values. This is how to assert that the degree is zero or negligible.

^:i !jw |kseo !fu /gr -ksi

This cheese is *not* fresh. When the dimension ranges from positive to negative values, **gr** interchanges positive and negative. On occasion, **gl** will also apply to indicate the zero point, though it is meaningless with "ksi-*fresh"*. For extremes of unfreshness one can use "fpu-*rotten"*.

^:i !jw |kseo /fi -vry -can -psl

This cheese is *de*solidifying. When a process occurs in the reverse of the usual order, "vry-*reverse"* indicates this.

De Morgan's Rules in Quantification

Negation interacts with *"and"* and *"or"*, which necessarily occur in sentences which are quantified or whose arguments have multiple referents. Therefore it is advisable to digress into some elementary symbolic logic. Here is De Morgan's rule for negation, stated four ways: (A and B are sentences)

A *and*

B = *not(*

(not A) *or*

(not B)) *(not A)*

and *(not B)*

= **not(** *A*

or *B)*

A *or*

B = *not(*

(not A) *and*

(not B)) *(not A)*

or *(not B)*

= **not(** *A*

and *B)*

Remember that in logic, *"A or B"* is true if one or *both* of the statements is true, unlike in English where the *"or"* generally excludes both being true.

Universal quantification means a statement is true when applied to all members of a set, of the form *"S1 and S2 and S3 and . . ."*, where S1 is the statement applied to member 1 and so on. Existential quantification means that a statement is true about at least one set member, in form *"S1 or S2 or S3 or . . ."* When such statements are negated, De Morgan's rule applies. Here are some more specific examples.

^:i -kio !ji ^kseo
I have the cheese. This will be the basic example sentence. Let us make the existential quantification more explicit:

^:i -kio !ji ^kseo |zu -to
I have at least one piece of cheese. Existential quantification like this means the same as *"I have piece 1* or *I have piece 2* or *. . ."* for all pieces of cheese. Now the simplest negation of this sentence is simply:

^:i -sfa !kio !ji ^kseo |zu -to
It is false that I have at least one piece of cheese. This form does not suit typical speakers; we want to negate the relation word "kio-*possess*", not the whole sentence, like this:

^:i !ji /go -kio !xa -kseo
I don't have any cheese — I don't have piece 1 *and* I don't have piece 2 *and* . . . To negate (or de-negate) a disjunction (compound sentence with *"or"*), we had to change *"or"* to *"and"*, producing a universal quantification. The same principle applies when you start with a universal:

^:i !xa -xe -cil /jir
All the children are here — Child 1 is here *and* child 2 is here *and* . . . Rather than negating the whole sentence with "sfa-*false*", let us negate the predicate "jir-*here*":

^:i -go -jir !cil |zu -to
At least one of the children is not here — Child 1 is not here *or* child 2 is not here *or* . . . In general, when you negate the predicate of a sentence involving quantification or multiple argument referents of any kind, you will also have to reverse the type of quantification or conjunction used.

Sentence Forms

Moods and Imperatives

These are the mood prefixes in *gua\spi*, which indicate the manner of assertion of a phrase. A top level sentence has **ge** on it by default unless another mood prefix appears.

ge	Asserted to be real or factual	
	^:i !vo -ge -dae !kara !fu -bal !crw |jro ^tara ^kseo	If the box is open, *which it is,* then the rat could eat the cheese
gi	Potentially true; actual truth is irrelevant	
	^:i !vo -ge -dae !kara !fu -bal !gi -crw |jro ^tara ^kseo	If the box is open then the rat *could* eat the cheese
go	Unreal or counter to fact	
	^:i -go !ji /kio !tara |zey !ju	I *don't* have your rat
gu	Hypothetical; reality is irrelevant	
	^:i !ji /gu -fli ^:o -sar !gu -vlw !ji ^qyun	If I *could* fly I *would* go to the moon

Closely related to the mood prefixes is the aspect operator "tri-*ritual*", a sign of a performative phrase. "*Performative*" means that by uttering the words the speaker makes something true, as in a marriage vow or the illustrated naming ceremony. Note that auto-conversion is suppressed by **zo**; without it, the sentence would merely be the topic of a ceremony, not the ceremony itself.

> **^:i \|zo -tri ^qo -ben /zu -xim !jw \|cil** *(Performative:) Ben is the name of this child*

In English there is an imperative mood; however, in *gua\spi* you make a sentence imperative by using "jo-*you*" or "ja-*we*" in the case for the actor, generally the first. These pronouns are distinguished from the non-imperative "ju-*you*" and "je-*we*". A decoration "pli-*please*" softens the command. For example,

> **^:i \|faw ^vu -qnu !qo -josefo /jo /qma -duw !gunu !ju** *Josepho, move your ass!*

> **^:i \|vi -pli ^jo /pin -dwo** *Please be patient.*

Special Features of Infinitives

In an infinitive the previous argument is replicated by default as the infinitive's first argument, while the first argument of a subordinate clause comes normally from the restricted phrase. Hence numbered cases skip over the first argument, and you must use the caselink **so** for any explicit first case in an infinitive or subordinate clause. In an infinitive with **vo** a predicate is made out of the sub-sentence that follows, including arguments and clauses. In the rare case where a sub-phrase (like a subordinate clause) must go on the infinitive predicate rather than into the sub-sentence, you can put a prefix before **vo**, like an article, and put the clause between the article and **vo**.

When an infinitive with **vo** is an argument, the main sentence asserts the relation of arguments to the infinitive's events, but does not make a separate assertion of those events. To additionally assert or deny the sub-phrase, use **ge** or **go** respectively. For example:

^:i !qo -kira /juy -xna !do ^qnou !xgno *Kira allows it to hold his hand (offers — but instead it swims away — infinitive not asserted)*

^:i !do /qou !qo -kira ^ge -qma -za -pai !cana ^ve -tum =tuen *It watches as Kira bails (drains) the boat with a bucket (infinitive is also asserted)*

Comparative and Superlative Natural languages have various complicated arrangements to change a simple property to become comparative or superlative. *Gua\spi* does it with a predicate.

^:i !X1 /qaw -xgi !X2 *X1 is equally green as X2*

^:i !X1 /gre -xgi !X2 *X1 is more green than X2*

^:i !X1 /sym -xgi !X2 *X1 is (one of) the greenest member(s) of set X2*

In the case of *"sym-superlative"* it is possible for several members to be equally green, each being greener than the remaining members. Also, a numeric predicate modifying **sym** produces the N'th greenest member. Here are some sentences with comparatives and superlatives:

^:i !star -fn -siriu ^qo -prosyon /gre -xgm *The star Sirius is brighter than Procyon*

^:i !qo -jupiter /sym -kqa !stel *Jupiter is the largest planet*

**^:i !qo -siriu /sym -xgm |cu ^xu -star |vu -sen !zu -jrer \hfil
**
^:i !qo -siriu /fne |cu ^sty -xgm !xu -star *Sirius is the second brightest of all stars, as seen from Earth (two ways)*

^:i !qo -siriu /sym -xgm !tei !star ^qo -sol *Sirius is the brightest star except for the Sun*

Causal Sentence Connectives

^:i !tara /crw !kseo ^:o -kau !gai -tuol !kseo
The rat eats the cheese, and that *causes* it to be dirty. A cause is rather mechanical. Actors with free will are rarely caused to do anything, despite their protestations. Here the rat may have free will, but the cheese, caused to be dirty, certainly does not.

^:i !ji /gri !tara ^kai |kei ^kseo ^:o -kmo !qma -qtu !ji ^tara
I am angry at the rat for stealing the cheese, which *motivates* me to kill the rat. The theft motivates the anger and the

anger motivates the planned killing. When a free agent acts it is usually because of a motivation. Here the speaker includes *"*kei-*crime"* in the sentence as a justification for his action. The definition of this word reminds you that it has the modal case *"*tue-*culture"*, which presumably includes the speaker — but not the rat.

^:i !xi -tara /qai -crw |jro ^kseo ^:o -zu -zni !vel !klo ^kseo
So that rats cannot eat the cheese, is the *reason* the cheese is in a closed container. A reason is an end (ending event) or consequence that *motivates* someone to make a starting event happen, such as keeping the cheese in the box, that will *cause* the consequence. The concept of *"*zni-*reason"* is rather slippery. First, the desired or planned consequence should be stated, not its inverse; **^:i !xo -tara /gi -crw !kseo** = *"A rat might eat the cheese"* is the negative of the correct consequence. Second, we say in English *"past event Y is the reason for action Z"* where the *gua\spi* definition of *"*zni-*reason"* requires *"vengeance for past event Y"* — a future consequence of action Z. **vou** = *"vengeance"*. Third, a *"*gul-*rule"* can be said to *cause* its reason, provided the obligees obey it.

^:i -dae !kara ^:o -sny !pwo -cyr -xyn !xi -tara ^kara
The box being open *implies* that a rat can go into it. The relation of logical entailment has to do with definitions and theorems, not with the arrangement of the real world or the will of its actors. *"*zny-*imply"* is the corresponding set operator: *"X1 is the union of X2 and the complement of X3"*, where X2 and X3 can be infinitives with **vo**. Perhaps the distinction between **sny** and **zny** is merely an artifact of old *Loglan* and English usage. We shall see if this is true as *gua\spi* matures.

^:i -dae !kara ^:o -bal !crw |jro ^tara ^kseo
If the box is open *then maybe* the rat will eat the cheese. This kind of fuzzy inference based on real-world consequences is what people use most often, rather than pure logic.

^:i !ji /gu -fli ^:o -sar !gu -vlw !ji ^qyun
If I could fly I *would* go to the moon. Necessary conditions are very commonly expressed and the logical *"if-then"* catches their true meaning poorly. Related is *"*sno-*sufficient"*.

These are the sentence connectives most often seen. But the speaker may connect sentences with any useful word having suitable cases. And like all *gua\spi* words, the sentence connectives can also be useful as arguments and as modal caselinks.

Logical Sentence Connectives

Old *Loglan* was intended to be a *"logical language"*, thereby to differ as much as possible from English. Therefore, one of its key features is support for what amounts to spoken symbolic logic. This feature is de-emphasized in *gua\spi*; in practice, what language users encounter most often, and stumble over, are Cartesian expansion of multiple arguments, non-commutative quantification, and complicated negations. These topics are well-supported in *gua\spi*. Nonetheless, set arithmetic can be performed on infinitives and the result is a set of events to which the listener's attention is drawn, just as with a more normal sentence. The logician's *"if-then"* can be realized through **zny**. Here are some examples of logical sentence connectives:

> **^:i !xun !vo !ji /crw !ftu =plyw /vo !ji /crw !ftu =peir** *I eat an apple or I eat a pear (or both, per logic)*

> **^:i !ji /crw !ftu !xun !plyw ^peir** *I eat a fruit of the apple or pear tree (better sentence)*

> **^:i !zny !vo !xa -fma /zu -bor !cy /vo !fma /bor !jy** *If a shape has void boundary then it is itself the boundary of something*

Features of Thesaurus Categories

The *gua\spi* words have been put into groups with related meanings, for ease of learning and for ease of creation. The dictionary includes a thesaurus of these categories. Many categories have closely related cases, or certain special features, which are described below.

Abstract Comparisons

Many abstract comparisons (1.1.1) and set member words (1.1.3) include a dimension on which comparison occurs. In a compound with the dimension as sub-word, its cases merge in an unusual manner. Considering the dimension to be single-ended (e.g. a color, as opposed to a directional property), its first case is applied to several arguments as noted in the definitions, e.g.

> **^:i !X1 /qaw -xgi !X2** *X1 is equally green as X2*

> **^:i !X1 /gre -xgi !X2** *X1 is more green than X2*

> **^:i !X1 /sym -xgi !X2** *X1 is (one of) the greenest member(s) of set X2*

"xgi-*green*" is applied to both X1 and X2 in the first and second sentence. This is described as a *"dual merge"*. In the last sentence, "xgi-*green*" is applied to X1 and to members of X2. The dictionary indicates all these special merges.

"stl-*list*" involves a dimension which is applied pairwise to members of the list, indicating the ordering.

"qaw-*equally*" has a very unusual definition: the first case is an infinitive into whose first case the rest of the cases are copied in turn; the predicate means that all the arguments fit in the infinitive equally. Normally the predicate of this infinitive is provided by compounding, as in the example above.

Sets

For several words in category 1.1.2 (sets) of the form *"(set) X1 is a (whatever)"*, you can make a compound **vdr =W** to get the members.

When **xy** (in-mind set) is the default article for a case, then if the referents are sets the default changes automatically to **xe** (in-mind in extension). But **xu** (whole set) does not change to **xa** (same in extension) because in math functions the usual occupant of such a case is supposed to be a set of equal-count sets.

The predicates "tla-*set*" and "stl-*list*" have a special arrangement of cases. They mean *"X1 is a set (in extension) or list (ordered) consisting of members X2, X3, X4, . . ."*, as many cases as needed. If X2 etc. have multiple referents in extension (which must be ordered for **stl**), all referents go in the set or list. Five or six words have this *"as many as needed"* argument list.

Properties

Noncomparative Properties are distinguished in *Loglan* from the Comparative Properties in that it is not useful to say that X is more <whatever> than Y; for example, X is more dead than Y. For this reason *Loglan* Comparative Properties each have a case for the compared item and Noncomparative Properties do not. Nonetheless, many of its members may actually be used comparatively (like *"*ksu-*delicious"*) and the distinction is rather artificial. In *gua\spi*, Properties do not have comparative arguments.

Directional Properties (1.5): These are often compounded with motion words, in which the moving case is related to the destination. (Special case: *"*tai-*outside"* merges with the start point. Examples in *"Compound Words"*.) Note that the polarity (e.g. up/down) in such compounds is often backwards from English.

Timelike Directional Properties (1.5.3): These are the relation words for the tense modal case.

Behaviors

Abstract Behaviors(2.1): These have the form *"X1 does (vo) X2+1"*, in which X1 is automatically replicated as the first case of the infinitive **vo X2**.

Double Actor Transitive Activities (2.1.3): These have the prototype *"X1 makes X2 do (vo) X3+2"*, in which X2 is automatically replicated as the first case of **vo X3**.

Games for Two Players (2.1.4): Generally you will want to use a reciprocal construction like this, unless the relation really is unilateral:

> **^:i !qo -jan ^fe -qo -mery /kul !vr -zdmo** *John and Mary kissed each other*

Motion Words (2.2): The prototype is *"X1 goes to X2 (destination) from X3 (start point) via X4 (route)"*. Since motion words are complicated, effort has been put in to make them all regular. They are very frequently combined with directional properties, q.v.

Transitive Motion Words (2.2.3): The prototype is *"X1 makes X2 go to X3 from X4 via X5"*, and again they are all regular. Directional properties relate X2, the mover, with X3, the start point.

Quasi-motions and Routes (2.2.4): The routes are set up as regular motion words. The quasi-motions can profitably be compounded with motion words.

Communication and Mental Activity (2.3): The pattern *"X1 knows that X2 is (vo) X3+2"* is common, with X2 merging as the first case of X3. However, quite a few predicates in this category have different patterns, so watch out.

Transitive Activities with an Object (2.4): A number of these words have an X3 case for a tool or means which is typically filled by a transitive compound, as in:

> **^:i !ji /fey =cuem !kliw** *I pound on the nail (hammer hit)*

Things and Materials

Animals and Plants (3.1): These have just one argument. The animals and plants category has been extended to include a primitive for each phylum, or at least most of them.

Body Parts (3.2): These have the prototype *"X1 is a (part) of creature X2*"*.

Materials (3.3): Almost all of these are of the form *"(xo) X1 is a serving/portion of (material)"*. The **xo** appears by default when the word is used as an argument, unless the containing sentence provides a default article other than the usual **xe**.

Places, Seasons and Weather (3.5): Places mostly have the form *"X1 is a (place) of locality or superset X2"*.

Containers (4.1.1) and Cooking and Eating (4.1.2): These have the form *"X1 is a container containing (xo) X2*"*. Constructions like *"spoonful"* are handled with "ful-*contained quantity*", like this:

> ^**:i !ji /crw !ze -kme |ful =spun** *I take a spoonful of medicine*

Transport (4.1.4), Machines (4.1.5), and Parts of Structures (4.1.7): Many of these are like body parts: *"X1 is a (part) of structure X2*"*.

Houses (4.1.8): House parts are as above. Houses themselves have the form *"X1 is a (house) of resident X2*"*.

Cloth and Parts of Clothes (4.2.2): Parts are as above. Cloth has the form *"(xo) X1 is a portion of (cloth)"*.

Food (4.3): Mostly of the form *"(xo) X1 is a serving/portion of (food)"*.

Works of Art (4.4.1): All have the form *"X1 is a (thing) about X2 created by X3 and performed by X4"*. X2 may be an event or a thing; there is no **vo** default. X4 is present only on relevant words such as "jiul-*drama*".

Miscellaneous Categories

Nationalities (4.7.1): *Loglan* has words for nationalities, for the languages spoken there, and for the basis money unit of the nation. But only about fifteen arbitrarily chosen nations are supported, mainly European ones. *Gua\spi* uses foreign names for these concepts, through "zina-*nation*", "gua-*language*", and "cni-*money*". "spi-*person*" translates the usual self-referential word in primitive languages for ethnic members of that culture.

Business (4.7.3): A number of these words have the form *"X1 (sells) goods or services X2 to other trader X3 for amount of money (xu) X4"*.

Most Frequent Words

So far, the corpus of *gua\spi* text available for analysis consists of 3140 words of fiction representing a teenager setting up a small business and interacting with younger children, parents, customers and girlfriend. I originally wrote this story in *Loglan* to test various features, and it is known that word frequencies will differ in other topics. However, this text gives some guidance about which words a beginner should be sure to learn.

<TD do

Word	Count	Meaning
	Structure Words	
:i	259	Sentence start
zu	89	2nd case conversion
ql	55	Speaker \leftrightarrow listener
fi	43	Grammar to level zero
va	39	Subordinate assertion
sa	35	3rd caselink
qo	33	Foreign name
qa	31	Pop modal stack
fe	30	Conjunction
:a	29	Next sent in sequence
qe	28	Stack modal default
xo	28	Article ``any''
:e	26	Sentence conjunction
:o	24	Retroactive downjump
vo	23	Infinitive
za	21	3rd case conversion
gr	18	Linear negation
gl	17	Polar negation
xi	17	Article ``typical''
qi	14	Replace modal default
vi	14	Attitude indicator
fy	13	Retroactive downjump
vu	11	Restrictive clause
xa	11	Article ``all''
	Pronouns	
179	Variables	
ji	132	Me
ju	77	You

Grammar Algorithms

Backus-Naur Form For computer applications the normal way to represent *gua\spi* syntax is in a semi-procedural language such as Prolog, because the end of a phrase comes when the next word's level passes up to or above that of the phrase main word, rather than at a standardized ending word, and such a relation is hard to represent in BNF. However, it is generally expected that the syntax of a language will be presented in BNF, so here it is.

; Morphology.	
C	= (choice of letters)
Cseq	= (Cseq C) | C
V	= (choice of letters)
Vseq	= (Vseq V) | V
Word	= Cseq Vseq
; Tone categories.	
Compound	= -' | ='
Sametone	= `^'
Down1	= !' | |'
Up1	= `/'
; Grammar. LHS `-' symbol indicates which end has a tone.	
Prefix	= (subset of Word, e.g. *vo''</I> or <I>zu'*)
Primitive	= (subset of Word, e.g. *tara''</I> or <I>crw'*)
Phrase	= Prefix Args0 Phrase
<TD> | Phrase-w	
Phrase-w	= Primitive Phrase-w
<TD> | Primitive	
Phrase0-	= Phrase Sametone
<TD> | Phrase Down1 Args1	
; Args(n) is a list of phrases that jumps up n levels at the end. Args3, 4, . . . are defined similar to Args1 and 2. Some finite bound must be set on n to give a finite grammar.	
-Args0-	= Compound (Just one tone)
<TD> | Down1 Args1	
Args1-	= Phrase Up1
<TD> | Phrase Down1 Args2	
<TD>	

"Discourse" is the root grameme. Grammar for quoted non-*gua\spi* text is not shown, but foreign predicates and quoted *gua\spi* are processed by this grammar and are put together at the organizational syntax level.

Organizational Transformations

Formal syntax is finished at this point, and transformation begins, in this sequence of steps:

- Transform the tone | into *"*!vu-*subordinate clause"*.
- Do the transformations for retroactive downjumps and for error correction (*fa*).
- Distinguish arguments from sentences. Insert **!so -jy** as the placeholder for arguments' open first cases.
- Re-order argument lists according to caselinks and conversions. Insert placeholders for missing cases.
- Look up each word in the dictionary. Insert default articles, typically *"*xe-*the"*, before arguments. Insert default **vo**.
- In cases of compounding, replicate argument lists for parallel arguments; insert **vo** for compound infinitives; or demote a compound object into the argument list.
- Replicate main phrase arguments into infinitives.
- Substitute the antecedents for phrase-relative, modal and question pronouns. The antecedent of a question pronoun is found in the future answer.
- Deal with modal case stack operations.
- Insert modal case defaults in argument lists lacking them.
- Retrieve the referent sets for all words. From them, compute the referent subsets of arguments and of sentences.
- These are the relations being called to your program's attention. Update word referent sets accordingly, or take other appropriate action.

Conclusion

People developing applications in *gua\spi* need some assurance that the language is not going to shift out from under them; but *gua\spi* certainly did not arise perfectly formed from the brow of Athena. The originator of the language certainly wants a certain amount of freedom to tinker with his creation; but a significant reason for the limited popularity of *Loglan* has been that people are not sure what the language officially is and which way it will jump next. Therefore I am making this baseline duration commitment: there will be no major changes in *gua\spi* until 1 January 1991 (two years hence); until then the language described herein will be acceptable in the sense that software ought to be able to understand it, even if upgraded to handle minor revisions; and when the time of major revision comes, the changes will be made after consultation with the community of people actively using *gua\spi.*

I hope this brief introduction to *gua\spi* has whetted your appetite to learn more about it. As you have seen, it expresses typical human sentences easily and efficiently. But the meanings of the words, and particularly the meanings of the phrases and sentences made from them, are defined much more specifically and clearly than in even the best natural languages. Finally, and most significant for artifical intelligences, the resulting meanings are cast in a form that is ideal for modern fifth-generation languages — which, in fact, those languages were designed to represent. Thus the gap between human and machine languages is closed by *gua\spi.*

Bibliography

- [L1] Brown, James C. Loglan 1: A Logical Language. The Loglan Institute, Inc., Gainesville, Fla., 1975.

- [L4] Brown, James C. Loglan 4 5: A Loglan-English / English-Loglan Dictionary. The Loglan Institute, Inc., Gainesville, Fla., 1975.
- [NB2] Brown, James C. A Proposed Revision in the Structure of Loglan Words (Notebook No. 2). The Loglan Institute, Inc., Gainesville, Fla., 1982.
- [TL43] Parks-Clifford, J. Supplement to Loglan 1. \sl The Loglanist 4, 3 (Nov. 1980).
- [La] Information about *Loglan* may be obtained from The Loglan Institute, Inc., 1701 NE 75th Street, Gainesville, FL 32601.
- [Lja] A modernized version of *Loglan*, much closer to the original *Loglan* than *gua\spi*, is *Lojban*. Information is available from The Logical Languages Group, 2904 Beau Lane, Fairfax, VA 22031, or lojbab@lojban.org. For on-line access, send a message whose body is *"index lojban"* to listserv@hebrew.cc.columbia.edu.
- [Ga] Please contact the author at UCLA Department of Mathematics, Los Angeles, CA 90024-1555, or e-mail to jimc@math.ucla.edu.

Introduction to the Dictionary

James F. Carter

15 September 1991

This is the dictionary of the *gua\spi* language. It is organized into three parts in which words are ordered by *gua\spi* spelling (with morphological formats in separate sections), by the English translation, and by thesaurus categories. The table of contents also serves as an outline of the categories.

This edition includes only primitive words of *gua\spi*. Only a few compound words appear. A future project will be to go through a list of the N most common English words and either to verify that each one has a primitive translation or to recommend a compound representation of it. Note the word *"recommend"*. The speaker of *gua\spi* is expected to represent his meaning by compounding primitive words creatively, and the architect of *gua\spi* does not insist on particular compound words to translate particular English words.

But since the vocabulary of English is so vast, even the working vocabulary of an educated person, most English words will never be in this dictionary. Here is where the thesaurus can help. Look at the primitive words in the category where your meaning is, and try to modify one of them by compounding to achieve the meaning you want.

To get the most use out of *gua\spi* you have to know all the primitive words. For this the thesaurus can be helpful as it forms an organized structure whose lists are a convenient size for memorization.

Be sure to remember that *gua\spi* predicates can be converted. The English word used in the definition is the one most representative of the unconverted *gua\spi* case order, but by conversion you can get up to four additional English meanings.

In one trial of *Loglan* about half of the predicates were compound, and likely more will be compound in *gua\spi*. Be alert for creative opportunities for expression — don't use only the words identified as being *"common in compounds"*. But resist the temptation to specify a predicate over-precisely with many compounded words; one thinks differently in *gua\spi*, and mashing *gua\spi* words to fit English preconceptions does not give the best results.

Acronyms, so popular in illiterate English, are useless in *gua\spi* because the letterals of the acronym are longer than the compound word or phrase that they abbreviate. Again, keep the compound short.

An example of a definition is *"X1 likes to do (vo) X2+1"*. The symbols X1 and X2 represent the first and second numbered cases of the predicate *"like"*. Words in parentheses before the case symbols, such as *"(vo)"*, are provided by default before

that argument unless there is an explicit article there — which cancels both the default article and the default phrase linker if any. All arguments receive **xe** as a prefixed article by default, unless a different article is shown in the dictionary or is said explicitly. Signs following the case numbers indicate special features of compound words and case merging in infinitives. Here they are illustrated decorating the second case, but the same pattern applies for all cases.

X2*
In a transitive compound (tone =), this is the case where the object goes. It is the second case by default (but the dictionary even so shows many X2*'s explicitly).

X2+1
In an infinitive compound (tone -), this is the case where the infinitive goes. Note that a few words do not do infinitive compounds (they do parallel compounds instead) even though they have a case for an infinitive. All cases of these words lack `+' or have `@'. The `1' selects the first case of the main predicate. This argument is replicated as the first case of the infinitive (whether explicit or from a compound). The notation *"+1,2"* means that referents from both cases are replicated. A `+' alone signals infinitive compounding but with no case replication.

X2-3
Usually used with a `+' decoration, the –3 means that the third case of the main predicate is replicated as the second case of the infinitive.

X2+S1
The *'S'* means that, X1 normally being a set, its members are replicated in extension into the first case of the infinitive. *"-S1"* is also used for the infinitive second case.

X2+P1
The *'P'* means that the infinitive is applied pairwise to members of the first case set, as in sorting or finding an extreme member.

X2+1@
The `@' indicates that this case is not eligible for infinitive compounds, though the indicated replication occurs for explicit infinitives.

X2=
The = means that starting with the indicated case (here X2), all following cases have a similar role, as many cases as needed. The sum of several numbers, as many as needed, is a typical example.

X2?
The `?' indicates a special case replication, described after the definition.

Here are the types of compounds. See the reference manual for a complete explanation with examples.

Infinitive
The second word is the predicate of an infinitive in one of the first word's cases. (Tone -.)

Parallel
Both words share the same arguments so both relations are simultaneously true of each argument list. (Tone -, or **-fe** if an infinitive compound would supervene.)

Transitive
The second word is the predicate of an argument in one of the first word's cases, the second case unless otherwise noted. (Tone =.)

Here is a reminder of the non-English letters of *gua\spi*. See the reference manual for more examples.

Gua\spi	English	Examples of Pronunciation
c	ch	CHew, Ciao (Italian)
q	sh	SHoe
x	zh	aZure, breZHnev (Russian)
:	(pause)	the:apple, hawai:i (glottal stop)
#	uh	thE, Among (schwa)
y	i	knIt
i	i, ee	grEEn machIne (not eye)
e	e	bEd
w	ng	stroNG

Gua\spi Vocabulary Lookup

The original site used an HTML form backed by a CGI script. That lookup only works on the live server.

- **Live form:** jfcarter.net ~jimc/guaspi/xankua.html
- **CGI endpoint:** `https://www.jfcarter.net/~jimc/guaspi/xankua.cgi`
- **Machine-readable dictionary:** xankua.dat
- **CGI source (Perl):** xankuacgi.txt

Search hints from the original page: you may give several words (comma or space separated), or a thesaurus category such as `1.2.3`, or a partial category like `1.2` to list subcategories (no words).

TeX documents and miscellaneous

The following material was mirrored from the original `old/` directory (browse on the host).

Directory listing

| File | Size | | — | — | | `acmpaper.shy.html` | 374 bytes | | `attitudinals.html` | 2661 bytes | | `complaint.form.html` | 2650 bytes | | `cowan.msg.html` | 5927 bytes | | `design.txt` | 6823 bytes | | `eatapple.html` | 2837 bytes | | `gua2loj.txt` | 9269 bytes | | `guaspi.addresses.html` | 746 bytes | | `guaspi.sty` | 5051 bytes | | `guatex2html.html` | 14848 bytes | | latindic.zip | 400 KiB (binary) | | `Makefile.html` | 805 bytes | | `metfli.txt` | 23292 bytes | | `metflix.txt` | 27239 bytes | | `northwind.cantonese.html` | 5465 bytes | | `northwind.engl.html` | 3446 bytes | | `reif.txt` | 8352 bytes | | `short.msg.html` | 3600 bytes | | `tonal.natlangs.html` | 7185 bytes |

File contents (converted)

acmpaper.shy

```
% acmpaper.shy -- special macros for acmpaper.tex

% Punctuation in the story translations
L   0   <
R   0   >
SL  1   /%/
BS  1   sub { "\\" . $_[0] . "\\" }
PA  1   (%)
S   1   <SUB>%</SUB>

% One sentence from the story.
x   4   sub { &{$subst{exii}}() . &{$subst{ex}}(@_[0..1]) . &{$subst{endexii}}() . $_[2] . "<P>" .
$_[3]}

% special macros in the BNF grammar
shalign 1   <TABLE>%<TR><TD>
```

attitudinals

```
Return-Path: LOJBAN%CUVMA.BITNET@mvs.oac.ucla.edu
Received: by luna.math.ucla.edu
    (Sendmail 5.57/1.07) id AA02350; Tue, 12 May 92 16:29:18 -0700
Received: from mvs.oac.ucla.edu by julia.math.ucla.edu via SMTP
    (Sendmail 5.61/1.07) id AA09168; Tue, 12 May 92 16:29:18 -0700
Message-Id: <9205122329.AA09168@julia.math.ucla.edu>
Received: from UCLAMVS.BITNET by MVS.OAC.UCLA.EDU (IBM MVS SMTP V2R1)
   with BSMTP id 8304; Tue, 12 May 92 16:28:29 PST
Received: (from cmsa.Berkeley.EDU for <LOJBAN@CUVMA.BITNET> via BSMTP)
Received: (from MAILER@UCBCMSA for MAILER@UCLAMVS via NJE)
 (UCLA/Mail V1.500 M-RSCS5692-5692-62); Tue, 12 May 92 16:28:21 PDT
Received: by UCBCMSA (Mailer R2.08 R208004) id 5915;
          Tue, 12 May 92 16:27:57 PDT
Date:         Tue, 12 May 1992 19:04:25 +0000
Reply-To: Richard Kennaway <jrk@INFORMATION-SYSTEMS.EAST-ANGLIA.AC.UK>
Sender: Lojban list <LOJBAN%CUVMA.BITNET@mvs.oac.ucla.edu>
From: Richard Kennaway <jrk@INFORMATION-SYSTEMS.EAST-ANGLIA.AC.UK>
Subject:      Some attitudinals I would like to see
To: Jim Carter <jimc>



The experience of listening to some dire conference talks and
question-answer sessions leads me to suggest some attitudinals and
discursives that it would be useful to have words for:
```

Attitudinals typically used as one-word utterances when listening to someone else speaking:

"Your utterance is too vague for me to respond to."

"Your utterance is so vague, incoherent, and grammatically confused that it fails to communicate any meaning to me whatsoever."

"Please delete your utterance and try again."

"Your have not yet said enough for me to form a useful idea of your meaning, but I still have hope. Please continue."

"Your utterance makes too many unexamined assumptions for a simple agreement or disagreement to be possible."

Discursives and attitudinals (is there a difference?) for indicating the status of the different statements one may make in the course of a talk:

"As background which I expect you to be unfamiliar with, and which I will therefore briefly summarise..."

"As background which I expect you to be familiar with, but which I will state in order to indicate the context..."

"This is an important statement, for reasons which I will subsequently explain: ..."

"This is an important statement, for reasons which should be clear from what I have said already: ..."

"This is a frivolous aside: ..."

"The answer to your question will be quite lengthy. Please wait until it is complete before commenting further."

--
Richard Kennaway   SYS, University of East Anglia, Norwich NR4 7TJ, U.K.
Internet: jrk@sys.uea.ac.uk   uucp: ...mcsun!ukc!uea-sys!jrk

complaint.form

(Sent to a person who complained about Lojban...)

I'm the "proprietor" of the language -gua!spi, which is fairly closely related to Lojban, and I am working in the Voksigid design team, this being a language artifact with neo-latin-oid vocabulary and radical emphasis on cases, in contrast to Lojban's denial of their existence.

I am particularly interested to know what the convulsions were that gave you so much trouble, so that I can take mitigating steps in the design of these other languages. I don't want to bias your answer to fit my preconceptions, but I'll offer my representations of what people might

```
complain about in a language; you can pick one or more, or make your own.
Note that this is a "generic language complaint form" and I would hope
that quite a number of these complaints don't apply to Lojban (or Voksigid
or -gua!spi or...)

1.  It's just too hard for me to pronounce.

2.  It's harder than in other languages to learn the vocabulary; the
words sound alien; I have trouble to remember them; I'm always grepping
in the dictionary to locate words.

3.  In other languages I have a natural feel for which words go together
into phrases, whereas the phrase grouping signs in this language have
to be translated intellectually, which gets in the way of comprehension.

3a.  You misplace or forget one structure word and your phrase splatters
from end to end of the sentence.  The grammar is too touchy, nitpicky
or ungraceful in error recovery.

4.  I think of a familiar meaning and I can't find a word to represent
it; the words which are there have meanings that aren't helping me.

5.  In English and other languages there is a facility to stick basic
words together to get more precise or unique meanings.  I was not able
to locate the corresponding facility in this language, or was not able
to make it operate properly.

6.  My name, home town, favorite ethnic food, pet's species, etc. are
mangled in this language.  The machinery for importing foreign words
was hard to use or ineffective.

7.  Clauses, subsentences, infinitives and suchlike are poorly coordinated
with the main sentence.  I have to repeat a lot of main sentence phrases
in order to be sure who's doing what to whom in the subsentences, unlike
natural languages where these are usually handled automatically.

8.  The relations (predicates, verbs) in this language connect bizarre
arguments (nouns, cases).  For example I'm used to "actor causes event"
but you give me "event causes event", or "actor wants object" but you
have "actor wants event".  It's very hard for me to change my world
view to match what this language is dishing out.
```

cowan.msg

```
Return-Path: cbmvax!snark!cowan@uunet.UU.NET
Received: by luna.math.ucla.edu
    (Sendmail 5.57/1.07) id AA02333; Fri, 13 Dec 91 21:04:06 -0800
Received: from relay1.UU.NET by julia.math.ucla.edu via SMTP
    (Sendmail 5.61/1.07) id AA14686; Fri, 13 Dec 91 21:04:05 -0800
Received: from uunet.uu.net (via LOCALHOST.UU.NET) by relay1.UU.NET with SMTP
    (5.61/UUNET-internet-primary) id AA26218; Sat, 14 Dec 91 00:04:21 -0500
Received: from cbmvax.UUCP by uunet.uu.net with UUCP/RMAIL
    (queueing-rmail) id 000319.6682; Sat, 14 Dec 1991 00:03:19 EST
Received: by cbmvax.cbm.commodore.com (5.57/UUCP-Project/Commodore 2/8/91)
```

```
    id AA25140; Fri, 13 Dec 91 23:46:06 EST
Received: by snark.thyrsus.com (/\==/\ Smail3.1.21.1 #21.19)
    id <m0ksJZS-00016eC@snark.thyrsus.com>; Fri, 13 Dec 91 15:33 EST
Message-Id: <m0ksJZS-00016eC@snark.thyrsus.com>
From: cbmvax!snark.thyrsus.com!cowan@uunet.UU.NET (John Cowan)
Subject: A brief critique of -gua!spi
To: jimc (jim carter)
Date: Fri, 13 Dec 91 15:33:21 EST
X-Mailer: ELM [version 2.3 PL11]



The following is an unorganized series of notes commenting on -gua!spi
characteristics I have problems with.  Much of this, of course, reflects
Lojban biases.  (It's about time somebody ran your stunt the other way zo'o.)

Character set:  I find the use of c, j, q, x rebarbative.  In particular,
x = sh in some languages, notably Portuguese, and I doubt it has the
-gua!spi value in >any< language.  Likewise q = ch in Pinyin.

Phonology:  Affricates conflict with clusters; how can you reliably distinguish
between c and the cluster tq (resp. j and dx)?

Phonotactics (not explicitly discussed in the report):  I find the consonant
clusters way too loose: many of them are unreasonably difficult to pronounce.
I would suggest starting from the Lojban set (in -gua!spi notation:)

    qp qf qt qk
    sp sf st sk
    xb xv xd xg
    zb zv zd zg
    ts dz

and adding carefully and selectively.  What is the existing phonotactics
anyway?

Tones:   Six is too damn many.  Four (high, low, rising, falling) would be
more like it.  This would require a little more verbosity in subordinate
clauses and transitive compounds.  In addition, it would be useful if
every monosyllabic word had an alternative disyllabic version to make
rising and falling tones easier, i.e. no "VV" vs "V" contrasts where the
V's are identical.  (This may already be true, but should be guaranteed.)

Extensionality:  Defining predicates purely in terms of the result sets
has known problems.  In particular, "x1 has a heart" and "x1 has kidneys"
are true for the selfsame set of x1s, so in -gua!spi terms they are the
same predicate.  For mathematical predicates, this may be all right (to
be negative is the same as to be less than zero) but in the real world
it can lead to annoying results.  See Quine.

Macro-like pronouns:  These barf on self-referential sentences and also on the
Whoozis-Whatsis sentence (I forget the names of the guys who discovered it):

    The boy who deserves it will get the prize he wants.
```

Vanilla macroexpansion produces:

    The boy who deserves (the prize he wants) will get the prize
    (the boy who deserves it) wants.

and then:

    The boy who deserves (the prize (the boy who deserves it) wants)
    will get the prize (the boy who deserves (the prize he wants)) wants.

And so on. You lose.

Compounds: You don't give any examples in the report of compounds of length greater than 2. Transitive compounds and event compounds can both suffer from ambiguity about associativity.

Masses: As you know, I have problems with your identification of masses with sets. The set of rats is large if there are many rats, whereas (part of) the mass of rats is large if at least one rat is large. Masses may have properties that appear self-contradictory because they are the conjuncts of the properties of their components (the mass of you is green if at least one of your components is green). Sets, OTOH, have properties totally independent of the properties of their members.

Negation: I believe you need something equivalent to Lojban's "na'e" -- nonspecific scalar negation. "lo na'e gerku" is a non-dog, probably another kind of animal (or something else if a different scale is presumed). This is distinct from "lo na gerku" -- that which is not a dog, contradiction pure and simple. There also seems to be no way to express a metalinguistic negation (presupposition failure, categorical denial, etc.) like Lojban "na'i".

Parenthesis: It would be helpful to be able to have subordinate clauses which are marked as digressions -- () in English prose -- or editorial insertions -- [] in English prose.

Erase all: A sentence start word which erased all of the speaker's previous sentences (Lojban "su") would be useful.

Change of topic: Sentence start words to clear anaphora assignments (Lojban "da'o") and sticky tenses (approximately Lojban "ki" and/or "fu'o") would also be beneficial.

Subscripting: There should be a mechanism for attaching an arbitrary phrase (typically a number) as a subscript to any word without affecting the word's grammar otherwise. This is excellent for manufacturing extra variables and the like.

Emphasis: Lojban "ba'e" emphasizes the next word, bringing it to the speaker's particular attention (similar to moving it to the beginning or the end of its bridi). It is like italics in English prose.

Sentence anaphora: There seems to be no analogue of Lojban "di'u" and friends:

"the previous (following, earlier, later) utterance considered as an object".
This is distinct from "go'i" and friends, which repeat the utterance implicitly.

Crocodiles: "crocodile" is translated in the Report as "sper -fe -zgol", but
"zgol" is not in the dictionary. What does it mean? How about "kli"?

Elephants: was the Old Loglan for "elephant" really "dumbo" ? :-) Not in
Notebook 3.

--
cowan@snark.thyrsus.com   ...!uunet!cbmvax!snark!cowan
        e'osai ko sarji la lojban

design.txt

      General Comments on the Design of a Language
               jimc 930528

Before speaking, we conceive of "relations" between "things". One category
of "things" that we conceive of is physical objects; another is abstract
relations between such objects; yet another is the structures of speech
and thought themselves. Metaphysically defining the referents of the
above assertion is beyond the scope of this document; the scope consists
of describing what design features of a language are needed to make it
useful to humans for speech. Specifically, I am concerned with what is
called a "predicate language".

Before we speak, we conceive of a relation between things. We then say
words which are symbols by which a listener is supposed to be able to
identify, with more or less fidelity, the relation and the things which
we thought of. The relation is technically referred to as a predicate,
the things related are called arguments, and the placeholders in the
relation that can be occupied by arguments are called cases. You can
think of the predicate as a true-false valued function with a
particular number of cases. Another set of words for "cases" and
"arguments" is "formal parameters" vs. "actual parameters".

In Latin-derived natural languages the cases have names such as
"nominative" or "dative" and comparing one predicate to another there
is a certain similarity between the role in the relation of each of the
cases in different words. No claim is made here that such regularity
is found in every language or in all words of any particular language;
nonetheless such regularities help a designer or learner powerfully.

One way to define a predicate is to list every set of arguments that is
thus related. The list (a set of sets) is referred to as the "meaning
list" of the predicate. Some predicates (e.g. color words) have only
one case; for them the meaning list will consist of a set of sets each
with exactly one member (e.g. something with that color). This special
case is not so special after all, because vague are the rules by which
a language designer chooses which potential arguments in a relation
should get cases, and which should be brought in by other means; a
predicate may be designed with one case today and later be given more
of them, with no major philosophical effort.

Each predicate is represented by a word. A predicate word with its arguments (expressed by words) is called a "clause". Clauses have three roles in speech:

1. Connected discourse consists of a sequence of clauses, called sentences. The sentences contain all the other clauses in the speech. A sentence is the symbol set by which the speaker's meaning is presented to the listener. The predicate of the sentence is the relation which the speaker mainly wants the listener to be aware of. We say that the speaker "asserts" the sentence. Other varieties of sentences are questions and commands.

2. There are also supplementary subordinate clauses which have the same purpose as sentences, but which are buried syntactically. The speaker buries them either because they are less important, or because they have some special relation with some sub-clause of the main sentence.

In this analysis, emotional and discursive statements are considered to be sentences or supplementary clauses conveying to the listener the emotion of the speaker or hints about the organization of the discourse. In other languages they are analysed as a separate kind of clause (interjection, etc.), but I think they differ only in the topic being carried to the listener, not in the fact of conveying information or in the potential ways to represent the information by words.

3. Argument clauses are specialized to represent the arguments in a clause. Generally an argument clause represents a set of arguments, called its referent set. The argument clause is built from a predicate, and the referents are those meaning list elements which fit in a particular case of that predicate, consistent with whatever other verbiage is in the argument clause. For example, the referent set of "cat food" is the set of "things" A such that for some cat N, "eat(N,A)" is true.

4. Restrictive subordinate clauses have the job of retaining only a subset of a referent set or meaning list. For example, "favorite cat food" would be the same set as above, intersected with the set of all A which is favored (by presumably the particular cat N).

5. An abstraction represents the meaning list of some predicate (consistent with additional verbiage) interpreted as the referent set of an argument. This is how we talk about events and conditions.

The syntax of a language has two jobs. First, it must guide the listener to group the words into clauses; and second, it must enable him to recognize the roles of and relations between the clauses. Elements to be provided by the syntax include:

1. The role of the clause: main sentence, argument, abstraction, or supplementary or restrictive subordinate clause. Typically additional sub-categories of supplementary clause are found useful, e.g. for emotions. In many languages, commands, questions and suppositions are distinguished from assertions through machinery analogous to role

selection.

2. For arguments, the case it occupies in the containing clause. The case of an argument is a sub-type of its role as an argument.

3. The export case. Arguments and restrictive clauses carry out their role through a particular case of the predicate, which must be identified through syntax. Main sentences do not necessarily have a particular export case, since they don't export a referent set, although many languages distinguish one of the cases of a sentence. While supplementary clauses are functionally like sentences, in most languages they are represented syntactically similar to restrictive clauses and hence will have an export case. Similarly, abstractions don't necessarily have an export case, but surprisingly frequently they need to have some case of the containing clause patched through to one of their own cases, which is kind of the converse of exporting.

In addition in many languages, arguments bear a "determiner" which, besides syntactically "determining" the start of the argument, indicates transformations on its referent set. Typically seen transformations are:

1. In-mind vs. veridical: Do the referents really truly fit the predicate (so says the speaker), or is the predicate merely a suggestive cue to the listener? Example: "the woman was a man in drag": "woman" is in-mind whereas "man" is veridical. Variously stringent rules are seen for how to interpret in-mind arguments.

2. Sets vs. extensions: Should the referent set be interpreted as a set, or does the containing clause apply to its members one by one?

eatapple

To: lojban-list@snark.thyrsus.com
Subject: Usage of cleft places and raised sumti

I got translations of our current model sentences into several languages.
English:

I want to eat an apple.
I want an apple.
I try to open the door.
I try the door.

In Chinese, Hindi and Hebrew, "I try the door" is meaningless; in other words, this combination is not subject to sumti raising. However, "I want an apple" does mean something. For the versions which in Lojban are rendered with abstract sumti, 9 of 10 cases (counting English) are more or less clearly also rendered by explicit abstract sumti, though a tanru-like combination is also credible for some. In all of these, without any question at all in the informants' minds, x1 of the main predicate was at main level, but was also x1 of the abstraction. Thus is justified a rule in which x1 of the abstraction is copied from main level. In other words, these sentences are rendered with cleft places,

never uncleft.

Language results:

Latin (1st year student, not very authoritative):

```
    vello edere    malum
(I) want  (to) eat apple

    tempto aperire   portam
(I) try    (to) open door
```

Vello and tempto are members of a category of what they call "modal verbs" which expect an abstract sumti in x2. If I interpret correctly the book's explanation, "malum" is inside an explicit abstract sumti; it is not an argument of vello. Neither I nor my kid know enough Latin to tell whether or not "vello malum" or "tempto portam" mean anything.

Hindi:

```
me sabe  kana chahadi hum
I  apple eat  want    (verb marker)

muje sabe  chahia
I    apple want

mene darwaza kolne ki     koshesh ki
I    door    open  (glue) try     (glue)
```

I wonder if this is their style of parallel diklujvo, rather than a modal verb? My informant identified "ki" as a glue word. The explanation of "me sabe kana chahadi hum" sounded more like a modal verb.

"I try the door" is incomprehensible in Hindi.

Hebrew:

```
ani rotze le'exol  tapuax
I   want  to'eat   apple

ani rotze tapuax
I   want  apple

ani nissiti liftoax  et  hadelet
I   try     to'open  the door
```

"I try the door" is incomprehensible in Hebrew.

Chinese: Tone symbols: 1 - 2 / 3 | 4 ! 1-prime *. y = both schwa and "ue", sorry.

```
|wo !yao -cr /ping |guo
I   want eat apple fruit
```

```
|wo !yao -i  *gy          /ping |guo
I   want one (thing unit) apple fruit

|wo !xe -dze !cy -kai /men
I   try....  go  open door
```

"go" is the motion word. "go open" is a stereotyped parallel compound
that they use in this kind of phrase. Chinese has lots of these
stereotyped compounds, probably to add redundant information to aid in
resolving the meaning of the individual words.

```
|wo !xe -dze -tsy *gy          /men
I   try....  this (thing unit) door
```

This sentence is syntactically valid but is meaningless in Chinese.

        -- jimc

gua2loj.txt

                        What is -gua!spi

Keith Ericsson asked me to put together a short description of -gua!spi. In
writing about 20,000 words of Old Loglan prose, I validated the strengths of
Loglan but also encountered a number of severe useability problems (many of
which have been addressed in Lojban). The goal of Lojban, gloriously
accomplished, has been to finish Loglan and get it launched. But I felt
that the next step in evolution should be taken immediately. Here is a
technical description of -gua!spi and a small sample of machine-translated
text.

--Design Summary: Morphology. There are 11 V phonemes: aeiouylmnrw (y as in
knIt, w as in stroNG). There are 14 regular C's: bcdfgjkpqstvxz (c as in
CHew, q as in SHoe, x as in breZHnev, no h). A word (including foreign
words) is one or more C's and one or more V's. Each word also has a tone in
the Chinese manner, which cues grammatical structure. How simple this is!
One of the British logli, long ago, asked why we needed CCVCV words when we
had affixes, and I took his suggestion to heart.

--Grammar. There is only one kind of phrase consisting of prefixes
(e.g. conversions), pre-subphrases (usually arguments), the predicate
(possibly compound, i.e. more than one word), and post-subphrases. The
various roles of these phrases are not distinguished by the grammar -- a big
simplification. Rather, the phrases are classified in a subsequent
organization step. Falling tone (Mandarin #4, written !) cues the start of
a subphrase; rising tone (#2, /) returns from it; and high-even tone (#1, -)
connects words at the same level. Up-down (^) starts another subphrase at
the same level; down-up (#3, |) marks a subordinate clause; and low-even (=)
is for a transitive argument. Multi-level upjumps are possible but are
needed rarely. A non-extensive experiment shows that naive non-Chinese
listeners can hear the tones.

--Organization. Caselinks and conversions are removed and arguments are

assigned to cases (X1, X2...) Infinitives (X1 is an event of [sentence]) start out either with the infinitive marker -vo as a prefix or as a compound word (i.e. -can-xyn = changes to be inside = enters); the subsentence is pulled out as a separate subphrase and -vo becomes the main phrase predicate. Extensive defaults are provided, of which -vo on X2 of -can is an obvious example. The definition of -can says that its X1 is replicated as X1 of its infinitive X2 so the actor which changes is also the thing that is inside. (Loglan never had this, and Lojban could use it.) There are similar services for parallel compounds like -bil-fli = fly from below = ascend. The extensive defaults and transformations make -gua!spi quite a bit more compact than Loglan; experience shows that they are almost always needed, but when not, they can be evaded easily.

Any predicate in a subordinate clause is acting as a modal operator. Extending pc's compound tense idea, there is a stack for every modal case; dialog and relative tenses are handled this way. The stack holds the antecedents of modal (personal) pronouns. There is a complete set of phrase-relative pronouns but they are used less than in Loglan because, at the semantic level, an argument with -xe (the) and one predicate can reliably reach a prior argument using that predicate. Names, including variables, are recognized during organization and are processed like pronouns.

--Semantics. As in Loglan, predicates represent relations, that is, lists of thus-related objects (e.g. for "eat", my rat plus my cheese, and numerous other pairs of eater and food). Following pc's lead, I define a sentence to call the listener's attention to certain members of the main predicate's referent set, designated by the arguments.

--Vocabulary. The word list of Lojban has been swallowed almost in toto, and I have extended the scientific vocabulary in mathematics, chemistry, zoology, botany and agriculture. All your favorites are here: verbs of motion, causal connectives, directional properties, and so on. Nonetheless there are still only 1400 words, not counting structure words and letterals. I didn't get much out of the relation of Loglan words to natural language, and I had neither the skill nor the time to locate equivalents in eight or so languages (which the Lojban team did). Hence as word creation fodder I used only English, Chinese (Mandarin, from a native speaker) and Latin (as a surrogate for European languages). A trial with random assignments was unacceptably ugly.

In Loglan and Lojban there are many compound words with idiosyncratic definitions, and (most) multi-word predicates are interpreted through metaphor. In -gua!spi there are no special definitions for compound words; they are interpreted as multi-word predicates, and they are then transformed to infinitives, parallels and transitives by specific rules, and are then interpreted by the definitions of the individual primitives. Metaphor is available but is much less prominent than in Loglan.

For my first exercise in Loglan I translated a technical article, and I promptly and painfully felt the lack of mathematical expressions by which to express dimensioned quantities like "125 meters per second". In -gua!spi numbers are classes of equal-count sets, and therefore, with no special extensions whatever, the grammar supports mathematical expressions with both

Polish and infix notation in the guise of arguments that have arguments (all post-arguments or mixed pre and post).

--Progress and Plans. The most often asked question about -gua!spi is, can a language simplified to this degree actually support live human language behavior? Some time ago I wrote a substantial story in Loglan to investigate this question, and so far I have translated about 3300 words of it to -gua!spi. My correspondents suggest that I have a naive user translate it to English and see if the meaning comes through, but as only one user is available so far, this is not yet practical.

However, I have nearly completed a computer program to parse and organize -gua!spi text: everything above except semantics (which, yes, is planned eventually). Output can be a parse tree, or pseudo-English. A prime design goal of -gua!spi was that such machine analysis should be feasible, and this discipline has powerfully shaped the language. Here is a brief sample of the program's output, illustrating how much of the meaning can be recovered and how lively the story remains even after this kind of abuse. Digits are case numbers; \\ enclose the predicate of the phrase that a subphrase modifies; these phrases are in <>; and subsentences are in []. Pronoun antecedents are in ().

```
{^:i |qe-jai !qo -:mei-ci /vu-qi-qnu !qu-zgiw=xgri /vu-qnu !qu-zgiw=xgri
/vi-faw ^jw /pei-qmy !ji ^pqua}
{ Mei Chi exclaims: Tigereye, Tigereye, he dunked me! }
{ [this1 push1 i (:mei ci2) some water3] and [(:mei ci1) above2 (water2)]
  <speaker/say/ :mei ci1 \and2\> <listener/emphatic/ you (eye3) \and2\>} %%
{ "Dunk" is best translated "push from above" = -pei-qmy, which is
  transformed to two sentences in parallel.  English here has the advantage
  of a massive vocabulary of rarely used, specialized words.  "He" (sexist)
  comes out as "this".  Sorry, the program doesn't map "I" to "me".}

\x{^:i |vi-faw ^jo /pny !jw ^zgly }
{ Punish him for that! }{}
{ [imperative (eye1) punish <listener/emphatic/ you (eye3) \punish2\> this2
  (and3) <speaker/say/ :mei ci1 (punish2)>]} %%

\x{^:i |ql ^ju /pny !jw ^vu-gul !jo}
{ Tigereye: You have to punish him yourself. }{}
{ [you (:mei ci1) punish this2 <actor/must/ imperative (:mei ci1) \punish2\>
  <experiencer/attention/ (:mei ci1) (punish2)> <speaker/say/ (eye1)
  (punish2)>]} %%
{ |ql interchanges speaker and listener modal stack antecedents.  Stack modal
  cases are inserted on each sentence automatically. }

\x{^:i !jo /kuo |fw-xgri ^jw}
{ Nag him. }{}
{ [imperative (:mei ci1) talk <tiger \talk1\> this2 <experiencer/attention/
  (:mei ci1) (talk2)> <speaker/say/ (eye1) (talk2)>]} %%
{ "Tiger talk" is a metaphor for "nag", inspired by Chinese. }

\x{^:e !jo /pei-qmy !jw |va-din /pqua }
{ (Or) dunk him back. }{}
{ and [imperative (:mei ci1) push1 this2 <exchange \this1\> some water3]
```

```
  and [(this1) above2 (water2)] <experiencer/attention/ (:mei ci1) (and2)>
  <speaker/say/ (eye1) (and2)>} %%

\x{^:i !jo /voi-zre !ji ^fyo !ju |zu-vem }
{ Don't ask me to fix you up when someone makes trouble for you. }
{ [imperative (:mei ci1) avoid to2 [(:mei ci1) request2 i (eye2) to3 [(eye1)
  repair2 :mei ci2 <victim/trouble/ \:mei ci2\>]] <experiencer/attention/
  (:mei ci1) (avoid2)> <speaker/say/ (eye1) (avoid2)>]} %%
{ Extensive reliance on defaults and replicated tenses to make three nested
  infinitives and one subordinate clause mean something.  Here -gua!spi
  clearly is more compact than English. }

\x{^:i |qa-jai ^qo-:mei-ci /gr-pli !do ^bwu !xgno}
{ Narrator: Mei Chi is not pleased how he helped her. } {}
{ [opposite :mei ci1 please variable~b (eye2) to3 [(eye1) help2
  (:mei ci2)] <author/fiction/ kar tr jym3 (please1)>]} %%
{ The narrator, Kartr Jym, sat on the stack all this time.  !do, a "variable"
  used as a name, was earlier assigned to Tigereye. }
```

guaspi.addresses

```
To:       lee@uhccux.uhcc.Hawaii.Edu,
        Gordon_Greene@MTS.RPI.EDU,
        jrk@information-systems.east-anglia.ac.uk,
        <cowan@snark.thyrsus.com> "John Cowan"
        hermix!kenf@anes.ucla.edu,
        JBALTZ@NEVIS.BITNET,
        <ftslr@acad3.alaska.edu> "Steve Rice",
        <fschulz@pyramid.com> "Frank Schultz",
        <71750.2413@compuserve.com> "Russell Whitaker",
        <chalmers@violet.berkeley.edu> "John H. Chalmers Jr.",
        <bob@gnu.ai.mit.edu> "Bob Chassell",
        <avl0@gte.com> "Alan Lemmon",
        <sunderme@stolaf.edu> "Brian A Sundermeier",
        <laibow@brick.purchase.edu> (Marnen Laibow-Koser),
        <allan.bailey@tamu.edu> "Allan Bailey",
        <cbogart@csn.org> "Chris Bogart",
        <jsp@betz.biostr.washington.edu> "Jeff Prothero",
        <s_nickn@EDUSERV.ITS.UNIMELB.EDU.AU> "Nick Nicholas",
```

guaspi.sty

```
% guaspi.sty -- Macros for -gua\spi text.  OK with LaTeX, probably AMSTeX too.

\parskip=\baselineskip \divide\parskip by 2
    \advance\parskip by 0pt plus 4pt minus 2pt
\let\guafont=\sl    % The font used in the \gua environment

% Special symbols -- these are for use outside \gua
\def\qh{\hbox{`-'}}     % Quoted hyphen
\def\!{$\backslash$}    % Backslash
\catcode`\|=\active%
\def|{$\vert$}      % Vertical bar
```

```
\catcode`\^=\active % Wedge (caret) is printing.  Use \sp for superscript.
\def\caret{${}\sp\wedge$}    % Caret without kerns for slanted type
\newbox\tonewedge \setbox\tonewedge=\hbox{\guafont\/}
    \setbox\tonewedge=\hbox{\kern\wd\tonewedge\caret\kern-\wd\tonewedge}
\def^{\hskip0pt\copy\tonewedge} % This caret is kerned over for slanted type.
\newdimen\SEkern \SEkern=0.16em
\def\dotSE{.\kern\SEkern.\kern\SEkern\@.\space}

% Environment for running -gua\spi text.  It's necessary to suppress line
% breaks after a hyphen (tone symbol).

\newbox\emgbox
\def\guaemg#1{\setbox\emgbox=\hbox{\strut #1}\hbox{\vtop{\copy\emgbox%
\vskip-0.5ex\hrule height0.25pt width\wd\emgbox}}}
\newbox\ghyphbox \setbox\ghyphbox=\hbox{\guafont -}
\def\guahyph{\discretionary{}{\copy\ghyphbox}{\copy\ghyphbox}}
\newbox\ehyphbox \setbox\ehyphbox=\hbox{-}
\def\englhyph{\discretionary{\copy\ehyphbox}{}{\copy\ehyphbox}}
\newbox\emdashbox \setbox\emdashbox=\hbox{---}  %No ligature with active hyphen
\def\emdash{\copy\emdashbox}
\let\=\-  %Since \- cannot be recognized with active hyphen

{\catcode`\-=\active%
\global\def\gua#1{%
{\guafont \hyphenchar\font=128%
\catcode`\-=\active%
\let-=\guahyph%
\let\emg=\guaemg%
#1}}
}

\def\qgua#1{\gua{``#1''\/}}             % Quoted gua\spi word
\def\guaspi{\gua{gua!spi}}              %The name of gua\spi
\def\Guaspi{\gua{Gua!spi}}              %The name of gua\spi capitalized
\def\trw-#1,#2,{``\gua{#1}-\hskip0pt#2''}   %A word with its translation

\def\emg#1{{\em #1\/}}          %A word in italics (underline in \gua)
\def\betw#1{$\langle$#1$\rangle$}       %A phrase in angle brackets < >
\def\hfilbreak{\hfil\penalty800\hfilneg}  %Allow break at awkward place
\def\afilbreak{\vfil\penalty400\vfilneg}  %Similar for vertical break
\def\vabreak{\noalign{\afilbreak}}  %Grudging break in halign

% Macros for alignments and paragraphs
\newif\ifphalign \phalignfalse
\def\phalign{\ifphalign\message{Wrongly nested phalign}\fi\phaligntrue
\begingroup\vskip\parskip\afilbreak
\everycr={\noalign{\penalty800}}\halign}
\def\endphalign{\endgroup\ifphalign\else\message{Extra endphalign}\fi
    \phalignfalse\vfil\penalty300\vfilneg}
\def\shalign{\vskip\parskip\halign} %halign with parskip before
\def\vhalign#1{\vskip\parskip\vbox{\halign{#1}}}    %halign in a vbox
\def\littlepar#1#2{\vtop{\hsize=#1\noindent%
\tolerance=1000\finalhyphendemerits=500\adjdemerits=500\doublehyphendemerits=500%
\strut#2\strut\par}}
```

```
% 2-column examples
\newdimen\exleft   \newdimen\exright
\newdimen\exindent \exindent=0pt \newdimen\exskew \exskew=0pt
\def\exarith{\exleft=\hsize \advance\exleft by -\exindent
     \advance\exleft by -1.5em \divide\exleft by 2 \exright=\exleft
     \advance\exleft by \exskew\advance \exright by -\exskew}
{\catcode`\-=\active
% \newenvironment{exii}         %A sequence of examples
\global\def\exii{\vskip\parskip\exarith\relax\catcode`\-=\active\let-=\englhyph}%
\global\def\endexii{}%
}
\def\ex#1#2{\penalty800\hbox to \hsize{\hskip\exindent %One example in a list
    \littlepar\exleft{\gua{#1}}\hfil\littlepar\exright{#2}}}
\def\pli#1#2{\begin{exii}\ex{#1}{#2}\end{exii}}  %A single 2-column example
\def\exbox#1{\vskip\parskip\vbox{\exarith#1}}   %Examples in a vbox

% Format for word lists, like this: \begin{exii} \xitemarith{wordsize, 4em}
% The items are xitem{word}{description}<NL>{-gua!spi example}{translation}
\def\xitemarith#1{
    \exindent=#1 \exarith
    \newdimen\y \y=\hsize \advance\y by -\exindent
}
\def\xitem#1#2#3#4{\vbox{
    \hbox{\hbox to \exindent{\gua{#1}\hfil}\littlepar\y{#2}}\ex{#3}{#4}}}

% Page style "dict" (\pagestyle{dict}).  The explicit marking is necessary
% because LaTeX uses \firstmark whereas for the dictionary it is necessary
% to use \topmark.  Also, the page number is wanted and LaTeX gives no access
% to the foot line.  The text should put out \mark{{text of left head}{right
% head}}.
\newtoks\chaptertok \chaptertok={}
\newtoks\sectiontok \sectiontok={}
\def\dictmark#1{\mark{{\sl#1\hfil\the\chaptertok}{\sl\the\sectiontok\hfil#1}}}
\def\ps@dict{                   %Activation of dict pagestyle
\def\@mkboth##1##2{}
\def\@evenhead{\expandafter\@leftmark\topmark}%
\def\@evenfoot{\rm\hfil\thepage\hfil}%
\def\@oddhead{\expandafter\@rightmark\botmark}%
\let\@oddfoot=\@evenfoot%
\def\chaptermark##1{\chaptertok={##1}\sectiontok={}\dictmark{}}
\def\sectionmark##1{}
\def\subsectionmark##1{}
\def\subsubsectionmark##1{}
}


\def\guachapter{\@startsection {chapter}{1}{\z@}{-3.5ex plus -1ex minus
    -.2ex}{2.3ex plus .2ex}{\huge\bf}}

\hyphenation{whe-ther re-fri-ger-a-tor me-ta-phor me-ta-phors
    au-stra-lo-pi-the-cus au-stra-lo-pi-the-cine
    ne-gate ne-gated ne-gat-ive sur-prise mod-al }
```

guatex2html

```
#!/usr/local/bin/perl
# guatex2html -- Translate the -gua!spi papers into HTML.
# Usage: guatex2html file.tex > file.html
# BEWARE: This program will handle *most* constructions, but further hand work
# is needed on the hard parts.  So don't blindly rebuild an existing HTML file
# as you'll ruin the hand labor.  Items to look for:
#   * Macros of the "halign" type are not handled at all.  You need to
#   turn them into proper HTML tables.  Also, \obeylines can't be obeyed.
#   * \xitem is defined differently in different files and tables within each
#   file.  Have fun.
#   * In some cases, 2-column tables looked a lot better with 3 columns,
#   like this:
#       <tr><td>Word <td colspan=2>What it means
#       <tr><td><td>-gua\spi example <td>Translation
#   It would have saved a lot of work to put this definition into a local
#   file where it's relevant:
#       xitem 3 <tr><td><i>%</i> <td colspan=2>% <tr><td><td><i>%</i> <td>%
#   * \def is programmed to eat everything up to and including the {}, but
#   about three \defs have highly deceptive nesting behavior, so that
#   sections of wanted text got eaten -- half the file, in the worst case.
#   * In a few cases the TeX files use \emg to italicize Roman text.  This is
#   bogus, and this program will underline it.

# If the command line file is file.tex, after the standard macro expansions
# are loaded, the program will read file.shy if it exists, to supplement the
# macros.  See \xitem above for the format; the args go directly to &addhash.

die "file.tex (exactly one) is required\n" unless @ARGV == 1;
die "No file $ARGV[0]\n" unless -r $ARGV[0];

$guafont="<I>";       # The font used in the \gua environment
$guafonte="</I>";

# Adds key-value pairs to %subst without replacing what's there.  Args:
#   $key        Name of macro without leading backslash
#   $narg       Number of arguments of this macro followed by the
#           separator (if any).  All separators must be the same.
#   $value      Either a string or a code ref.  For a string, each
#           successive % is replaced by one argument.  If a code
#           ref, the arguments are the macro args and the return
#           value is what to replace them with.
# Arbitrarily many triplets are given.
sub addhash {
    my($k, $n, $v);
    while (($k, $n, $v) = splice(@_, 0, 3)) {
    $n =~ /^(\d*)(\D*)$/;   # Number of args + separator, e.g. "2,"
    $nargs{$k} = $1;
    $sepr{$k} = $2;     # Usually this is ''
    unless (defined($v)) {
        die "Error, args to addhash are out of sync.  Keys:\n`",
        join("', `", keys %subst), "'\n";
    }
```

```
        # Now convert $v to a subroutine as described above.
        # (It could already be a subroutine.)
    if (ref($v) eq '') {
        my(@parts) = split('%', $v, -1);    #Don't lose trailing null fields
        my $w = shift @parts;       #Eventual return value
        my $i = 0;
        foreach $_ (@parts) {
        if (substr($w,-1) eq "\\") {    #Don't replace \%
            substr($w,-1) = "%$_";
        } else {
            $w .= '$_[' . $i++ . ']' . $_;  # % -> $_[$i]
        }
        }
        $w =~ s/(?=[\\"])/\\/g;     #Put \ before metachars
        my $cmd = "sub { \"$w\" }";     #Make the subroutine.
        $v = eval $cmd or warn "Key $k $cmd --- $@\n";
    }
    $subst{$k} = $v;
    }
}

our %subst;     # Macros, 0 or 1 argument.  % represents the arg.
            # The key is the macro name without leading backslash.

&addhash(
# Escaped characters
    ' ' => 0, " ",       # Escaped space '\ '
    "\n"    => 0, "\n",      # Accidentally escaped newline "\\\n"
    '/' => 0, "",        # Italic correction (ignore it) '\/'
    '-' => 0, "",        # Hyphenation hint (ignore)
    "=" => 0, "",        # Alternative to \- when hypen is active
    '%' => 0, "\\%",     # Escaped percent
    '#' => 0, "#",       # Escaped pound sign
    '*' => 0, "*",       # Escaped asterisk (guaspi special)
# Active characters
    '~' => 0, " ",      # Nonbreak space
    '&' => 0, "<TD>",      # Separator in halign
    '
    => 0, "",        # Math mode (ignore it)
    '{' => 1, "%",       # Grouping characters
    '}' => 0, "",        # Grouping characters
    '[' => 1, "[%",      # Visible grouping characters
    ']' => 0, "]",       # Visible grouping characters


# General TeX and LaTeX definitions
    title   => 1, "<H1 align=center>%</H1>",
    author  => 1, "<H3 align=center>%</H3>",
    date    => 1, "<div align=center>%</div>",
    abstract    => 0, "<blockquote>Abstract: ",
    endabstract => 0, "</blockquote>",
    chapter     => 1, "<H1 align=center>%</H1>",
    section     => 1, "<H2 align=center>%</H2>",
    subsection  => 1, "<H3 align=center>%</H3>",
```

```
    appendix    => 1, "<H2 align=center>Appendix: %</H2>",
    label   => 1, "<A name=\"%\"> </A>",
    it  => 1, "<EM>%</EM>",
    langle  => 0, "<",
    rangle  => 0, ">",
    P   => 0, "¶",
    quad    => 0, "    ",    #Disgusting
    qquad   => 0, "        ",
    itemize => 0, sub { $subst{item} = sub { "<LI>" }; $nargs{item} = 0; "<UL>"; },
    enditemize  => 0, "</UL>",
    list    => 0, sub { $subst{item} = sub { "<DT>$_[0]<DD>" }; $nargs{item} = 1; "<DL>"; },
    endlist => 0, "</DL>",
    cite    => 1, "[%]",
    sp  => 1, "<SUP>%</SUP>",    # Superscript
    sc  => 1, '<SPAN style="font-variant:small-caps">%</SPAN>',
    ref => 1, "[%]",     # Not really functional

# It's assumed that all tables have the form
# \begin{table} \halign{stuff} \caption stuff \end{table}.
    table   => 1, "<HR><TABLE>%",
    halign  => 1, "<TR><TD>%",
    cr  => 0, "<TR><TD>",    # Spurious row at end of table, too bad
    caption => 1, "</TABLE>\n<BLOCKQUOTE>",  # Lose [title of table]
    endtable => 0, "</BLOCKQUOTE><HR>",
    figure  => 0, "<HR>",      # Can't do much with a figure
    endfigure => 0, "</BLOCKQUOTE><HR>",
    phalign     => 0, '<BLOCKQUOTE><TABLE WIDTH="100\\%"><TR><TD>',
    endphalign  => 0, '</TABLE></BLOCKQUOTE>',
    vhalign     => 1, '<BLOCKQUOTE><TABLE WIDTH="100\\%"><TR><TD>%</TABLE></BLOCKQUOTE>',

    noindent => 0, "",       # Can't turn indentation on or off
    penalty => 1, "",        # Ignore penalties.
    def => 1, "",        # Ignore TeX macro definitions.
    newcommand  => 3, "",    # Ignore LaTeX macro definitions
    renewcommand    => 3, "",    # Ignore LaTeX macro definitions
    documentstyle   => 2, "",    # Ignore various LaTeX admin stuff
    oddsidemargin   => 2, "",
    evensidemargin  => 2, "",
    document    => 0, "",
    enddocument => 0, "",
    maketitle   => 0, "",
    protect     => 0, "",
    vskip       => 1, "",

# Definitions from guaspi.sty
    qh  => 0, "`-'",     # Quoted hyphen
    '!' => 0, "\\",      # Backslash
    '|' => 0, "|",       # Vertical bar
    caret   => 0, "^",       # Caret without kerns for unslanted type
    dotSE   => 0, " . . .",   # Ellipsis dots...
    dots    => 0, " . . .",   # Ellipsis dots...

# Environment for running -gua\spi text.  It's necessary to suppress line
# breaks after a hyphen (tone symbol).
```

```
    guaemg  => 1, "<blockquote>$guafont%$guafonte</blockquote><rule>", #The rule is in the original
but it seems bogus.

    guahyph => 0, "$guafont-$guafonte",
    englhyph => 0, "-",
    emdash  => 0, "---", #Need an em-dash by cowboy programming
# An inline word or short phrase in -gua!spi
    gua => 1, "$guafont%$guafonte",
    qgua    => 1, "$guafont``%''$guafonte",  # Quoted gua\spi word
    guaspi  => 0, "${guafont}gua\\spi$guafonte", #The name of gua\spi
    Guaspi  => 0, "${guafont}Gua\\spi$guafonte", #Same, capitalized

# A word with its translation.  Format: \trw-gua,english,
    trw => "2,", sub { "``$guafont" . substr($_[0],1) . "$guafonte-" . $_[1] . "''" },

    emg => 1, "<U>%</U>",    # An emphasized gua\spi word, underlined
    betw    => 1, "<%>",   # A phrase in angle brackets < >
    hfilbreak   => 0, "",    # Ignore various line break adjustments
    afilbreak   => 0, "",
    vabreak     => 0, "",

# These names pertain to \halign: phalign endphalign shalign vhalign

# Paragraph in table cell.  It's used with 2 arguments: a width and the
# content.  Toss the width, leaving the content.  Return value: empty string.
    littlepar   => 1, "",

# 2-column examples.  exii is seen as \begin{exii} and endexii is \end{exii}.
    exii    => 0, '<blockquote><table width="100\\%"><col width="50\\%"><col width="50\\%">',
    endexii => 0, "</table></blockquote>",
    ex  => 2, "<tr><td>$guafont%$guafonte<td>%",
# A single 2-column example
    pli => 2, sub { &{$subst{exii}}() . &{$subst{ex}}(@_) . &{$subst{endexii}}() },
# A lot of 2-column examples
    exbox   => 1, sub { &{$subst{exii}}() . $_[0] . &{$subst{endexii}}() },
# Word lists have two 2-column lines.  Args are:
# \xitem{word}{description}<NL>{-gua!spi example}{translation}
#   xitem   => 4, sub { &{$subst{ex}}("** $_[0]", $_[1]) . &{$subst{ex}}(@_[2..3]) },


);  # End of loading %subst

# Load an auxiliary definition file if present.  Its lines have the format:
#   macroname nargs content
# with fields separated by whitespace.  The content is the rest of the line.
# Blank lines and lines beginning with % are ignored.
if (($AUX = $ARGV[0]) =~ s/\.tex/.shy/ && open(AUX, $AUX)) {
    while (<AUX>) {
    next if /^\s*(%|$)/;
    chomp;
    my(@row) = split(' ', $_, 3);
    if ($row[2] =~ /^sub /) {
        my $cmd = eval $row[2];
        if (defined($cmd)) {
```

```
        $row[2] = $cmd;
        } else {
        warn "In $AUX $row[2] --- $@\n";
        $row[2] = " OOPS ";
        }
    }
    &addhash(@row);
    }
    close AUX;
}


# If an active character or macro name (without backslash) is a key in this
# table, its one argument extends to (and including) the macro (with backslash)
# or active character which is the value.
%endmarks = ( '{', '}', '[', ']',
    "def", '{',     # \def#1{whatever} this eats the {} too  \}
);

# These HTML objects are at block level and a <p> is not wanted before them.
%blocklevel = qw(<p 1 <ul 1 <ol 1 <dl 1 <li 1 <dt 1 <dd 1
    <table 1 <tr 1 <td 1 <th 1 <blockquote 1
    <h1 1 <h2 1 <h3 1 <h4 1 <h5 1 <h6 1 <pre 1 <div 1 );

# \hyphenation{whe-ther re-fri-ger-a-tor me-ta-phor me-ta-phors
#   au-stra-lo-pi-the-cus au-stra-lo-pi-the-cine
#   ne-gate ne-gated ne-gat-ive sur-prise mod-al }



# A TeX document is a list of tokens, which can be strings or sub-lists.
$bfr = join('', <>);      # Read entire document at once.
$bfr =~ s/(?<!\\)%[^\n]*\n//sg;  # Percents mark comments, to \n, which vanishes.
$j = 0;
TOKEN: {
    $j = &tokenize(\$bfr, undef, "\\enddocument", $j);
    last if $j >= length($bfr);
    &output(undef, "\n\n<br>==== Unbalanced right squiggle here ====<br>\n\n");
    redo;
}

$z = join("\n", sort keys %missing);
print  "\n<p>These macros have no definition:\n$z\n" if $z ne '';

# The active characters
BEGIN {
    %active = qw(\ 1 { 1 } 1 [ 1 ] 1 ~ 1 & 1 $ 1);
}

# Convert the buffer to tokens.  Args:
#   \$bfr   Ref. to linear input buffer
#   \$output    Ref. to linear output buffer, or undef for direct printing.
#   $end        The control sequence (with backslash) or active character at
#       which the unit ends.  It is included with the unit.  Specify
```

```
#       '' for exactly one token (or a subunit in { }).  Specify
#       \bye or \enddocument for the entire document.
#   $j      Index in buffer to start at
#   Returns:    Index in buffer just after $end
# It's assumed that no token can be over 100 bytes long.
sub tokenize {
    my($bfr, $output, $end, $j) = @_;
    my($h, $j0);
    my $len = length($bfr);
    $indent .= '*';         # Needed to know when to insert <p>
#   print STDERR "$indent Starting group `$end'\n"; #DEBUG
        # When hunting for macro arguments, whitespace before or
        # between arguments is skipped.
    $j += length($1) if substr($bfr, $j, 100) =~ /(^\s+)/s;
        # Split off tokens one by one.
    TOKENS: {
    last if $j >= $len;          # If end of input was reached
    $j0 = $j;               # Location of token start
        # Tokens consist of:
        #   % to end of line (comment, ignored)
        #   \alphabetic, a macro name, eating one space after
        #   digits followed by letters, a dimen
        #   word characters, a word
        #   a contiguous stretch of spaces including \n
        #   any single character.
    $h = substr($bfr, $j, 1);       #The next byte
#   print STDERR "`$active{$h}' ", &nonl(substr($bfr, $j, 10)), "\n"; #DEBUG
    if ($active{$h}) {
        $j++;
        my($sep);
        if ($h eq "\\") {           #A macro name
        substr($bfr, $j, 20) =~ /^([A-Za-z]+|.)(\s?)/s;
        $j += length($1);   # $1 = macro name
        $h = $1;
                        # Eat optional space after macro name, if there are
            # arguments, except leave a newline that doesn't
            # prevent recognition of args.
        $j += length($2) unless $nargs{$h} == 0
            || exists($endmarks{$h});
        }
        # Transform \begin{name} to \name, \end{name} to \endname
        if (($h eq "begin" || $h eq "end") &&
                substr($bfr, $j, 20) =~ /^\{(\w+)\}/) {
        $j += 2 + length($1);
        $h = (($h eq "end") ? $h : "") . $1;
        }
            # Extraction of arguments.  There are 3 styles:
        my(@args);
        my $na = $nargs{$h};
            # A special separator may delimit the argument(s).
        if ($sepr{$h} ne '') {
        @args = split($sepr{$h}, substr($bfr, $j, 100), $na+1);
        pop(@args);         #Lose text following special arg
        $j += length(join($sepr{$h}, @args, ''));
```

```
        } else {
            # Normally, a given number of ordinary tokens are used,
            # but a specific control sequence may be specified to
            # delimit the argument (generally only one).
        while ($na-- > 0) {
            push(@args, '');
            $j = &tokenize($bfr, \$args[-1], $endmarks{$h}, $j);
        }
        }
            # Do the macro substitution.
#       print STDERR "Macro sub `$h' args @args\n"; #DEBUG
        if (exists($subst{$h})) {
        &output($output, &{$subst{$h}}(@args));
        } else {
        &output($output, "\\$h" . join('', @args));
        $missing{"\\$h"}++;
        }

            # Numbers are special in TeX.  A dimension may follow.
    } elsif (substr($bfr, $j, 200) =~ /^(-?[0-9]+[a-z]*)/) {
        &output($output, $1);       #Ordinary text (to end of line)
        $j += length($1);

                        # Ordinary text includes letters, whitespace,
            # and certain punctuation not significant to TeX.
    } elsif (substr($bfr, $j, 200) =~ /^([A-Za-z.,;()`' \t]+\n?)/) {
        &output($output, $1);       #Ordinary text (to end of line)
        $j += length($1);
    } else {
        &output($output, $h);       #A single character
        $j++;
    }
    $needpara = length($indent)
        if (substr($bfr, $j-2, 3) =~ /^\n\n[^%\n]/);
#   printf STDERR "%-8s %s\n", $indent, &nonl(substr($bfr, $j0, $j-$j0)); #DEBUG
    } continue {
#   printf STDERR "end `%s' (%d) tail `%s'\n", $end, defined($end), &nonl(substr($bfr, $j0,
length($end))); #DEBUG
    redo unless substr($bfr, $j0, length($end)) eq $end;
    }
#   print STDERR "$indent Exiting group `$end'\n"; #DEBUG
    substr($indent,-1) = '';
    $j;
}

# Appends a fragment to the output stream.
#   \$output    Ref. to linear buffer for output, or undef for direct printing
#   $data   String to append
sub output {
    my($output, $data) = @_;
        # After an empty line in the input stream, insert a <p>,
        # except don't if the next HTML tag is at block level.
    if (length($indent) <= $needpara) {
    $needpara = 0;
```

```
    $data =~ /^(\<\w+)/; #Capture first HTML tag excluding arguments
    substr($data, 0, 0) = "<p>" unless $blocklevel{lc($1)};
    }
    if (defined($output)) {
    $output .= $data;
    } else {
    print $data;
    }
}

# Where a string has newlines, changes to "\\n".
sub nonl {
    my($data) = @_;
    $data =~ s/\n/\\n/sg;
    $data;
}
```

Makefile

```
# Makefile for producing the !gua-spi reference manual and dictionary.

all:    guatitle.dvi

# all:  guatitle.dvi wordchap.dvi englchap.dvi theschap.dvi

guatitle.dvi:   guatitle.tex guarefmn.tex dictintr.tex guaspi.tex guaspi.sty
    echo '\includeonly{guatitle,guarefmn,dictintr}' > selection.tex
    latex guaspi
    mv guaspi.dvi guatitle.dvi

wordchap.dvi:   wordchap.tex guaspi.tex guaspi.sty
    echo '\includeonly{wordchap}' > selection.tex
    latex guaspi
    mv guaspi.dvi wordchap.dvi

englchap.dvi:   englchap.tex guaspi.tex guaspi.sty
    echo '\includeonly{englchap}' > selection.tex
    latex guaspi
    mv guaspi.dvi englchap.dvi

theschap.dvi:   theschap.tex guaspi.tex guaspi.sty
    echo '\includeonly{theschap}' > selection.tex
    latex guaspi
    mv guaspi.dvi theschap.dvi

clean:
    rm -f *.dvi *.log *.aux *.toc selection.tex
```

metfli.txt

```
;
>ak%Q5
@H@Rla metflidjimao vedsia / Carter @W1,
```

@F@W1,@C@N
@B@C la metflidjimao vedsia
jia proza pui la kartyr. djim.
na 12/6/82
@W1,@E
@P.i la tigrnmenki ga katca lo valpu gi vi le kanla jua nu
proju lo junti
.i do fatru hirti lode ganta volsa jia karknpro ledo sorgu
.i lo lelpi kreni je la sol. jia pazi gangzo ga litndou
le cmalo po kamda vi lo cutri
.i cue la meicin. tei la tigrnmenki .i tei la tigrnmenki ta pa cao
damcuipuo mi .i kasfa ta
@P.i cue do tu kasfa oa ta
.i tigrnruu da .ica damcuipuo ta bai
.i noi raneu begco le po mi dremao letu nable
.i cue gu noi la meicin. ga nu pluci le po do pa helba
@P.i do zvosea ne lea nigro herfa ledo slopu menki
.i cue do jia penso gu nukou ie mi kecri
.i mi gudbi pui loe po fomcia
.i le ditca je mi pazi gudcue mi pei lemi cisti nu srite
.i cue gu do clidntco ledo fotli muslo .e ledo bilti mahgani skapi
gio jia roneu ce nona pluci do
.i cue do jia penso gu eu mi sucmi
.i cue do tei la .iciron. ea kadkukrgzo le neri grohaa
@P.i cue de aa .i hie le grohaa
@P.i cue do au .i cue gu de bortamgzo dio la .sol.
.i do favdjimao ledo rirgnkua .e ga renro da ne le bakso
gi vi le sucmrhao .e zi volti lo cutri
.i la .iciron. ga fotli sucmi .epou nu fotli ba jio deknie
.inokou do cmiza le po kadkukrgzo
@P.i ue le berbou cue do jia kraku .i nazi vi sucmi
.i tei la hosef. ga muvdo letu rirgu
.i ei tu nu titci ao le berbou pucpae
.i cue do jia penso gu uo mi katca oa ta nia ra lea minta
.i nukou ie ta selkerju noi
.i mi sucmi nia lo sekmi nera .ikou ta nurfatcea
@P.i cue ba mi picpro ao
@P.i cue do godzo .epou kukra .i cue gu le botci ga zvoselcpu lo cutri

@P.i cue gu le po la sol. ga gangzo ga drimao do le pu do tcicni
.i nukou ie cue do letau botci zu picpro
.i cue gu fazuna gu da favgzo
.i cue do na le midmia .i eo raba zvogzo .e resplicea
.i prigzo mi .i tei la glorias. letu lapla ga djifavcea tu
.i kamla .i muvdo .i noi pristo
.i cue gu do cpula ledo nu daspa ledo hasfa jia nu sitfa le feri
fordi je ne lea ganrorfamhaa
.i do hatmao lo tcidi tie le cmavau zavno
.i cue be jio cmalo gu mi tcicni .i eo kukra
@P.i cue do noi mi kukmro kokfa kao .i eo pispazda
.i cue za gu do donsu ra le junti lo tcidi
@P.i cue do jia sudna clacue gu
tei la meicin. tai nu tasgu mordu raba jio pa nu speni mi
.i godzo le krujao

.i cao nineu lia tau nurnitmao lotu tcidi
@P.i cue gu la meicin. ga godzo le krujao .e ga dridnkra
.i cue de mi tcicni .i mi muctcicni
@P.i cue do tu driki oe papa le po tu clidrkraco lotu tcidi
.i na santi
.i cue gu de prase dridnkra sui
@P.i za gu ra le junti ga kaldru le po da titci
.i la tigrnmenki ga tcicnisto .inurau kinumoi lo tcidi pa
sasirura tsufi raba ki do pa titci piri lo tcidi
.i do tokna leda pribadlo .e ga cutse ba gi tou
.i le grupa fa godzo le publai
.i le valda junti fa selji fomcia
.i cue da uouu
.i cue do jia norpoa pui toi gu tokna letu pribadlo .i ea na godzo
@P.i cue la rikin. jia fornie gu lo ctuda
@P.i cue do uo .i eo la .iciron. ga daspa .i mi ze la rikin.
fa vu jmite tu .i godzo ce durzo tei la rikin. .i mi fia pazda
[da is from the prev. paragraph.]
@P.i cue gu da ctupro .iza cue da eo klinrmao lemi gasno
@P.i cue do tu klinrmao oa le gasno je tu .i tu drudja tau
@P.i cue da .inorau tu pazda mi nurau ie
@P.i cue do klinrmao .i ei tu penso le pu oi tu tedcle
@P.i cue da mi mutce daspa
@P.i cue do ia tu daspa mordu loe fornie .ipou noi tu tedcle oi
.i ganmao letu rirgu .i ti tsuklini .i vlaci letu hanco
@P.i nana le po do ze da djicea le grupa gu la .iciron. ga fotli fadbei
.icepou da nu ge cnire dridnmenki .inurau la meicin. pia no kojdru da
.i cue da la meicin. pa tokna le tcaro pe la mzima .e drunortoi pui
le po favdou
@P.i cue la meicin. la mzima pa donsu ti mi
@P.i cue la mzima ia noi .i mi sira gruplilei lemi tcaro ta
.i tu zavlo ce solgrubro [no English equivalent]
rpui le pu tu tsitoa ba jio nu gruplilei
@P.i cue do ia .i donsu le tcaro la mzima
@P.i cue la meicin. uo noi .i tu pebtia ta .i ta donsu ti mi
.ice noi mi fa favdou ti
@P.i cue gu la tigrnmenki ga fadbrute [sigh] .e cutse ba gi tou
.i eo noi fatrdou .i gudbi durzo .i cue gu do filmo le po
la meicin. ga janjugpae [X is a trap to catch Y] fosmuvmao do
ba jio tsero
@P.i cue de tu pebtia .e pebtia .e pebtia .i mi tsodi tu
.i cue gu de bloda do le tcaro jia kutla ledo barma
.i duonoi le po penso gu do hanblo lede fasli
.i de kraku ba gi tou .i tu zavlo .i tu pa surna mi
.i tu favdaspa .i pae pae pae
@P.i cue gu le po za kasfa ga clado .e langa .ice nafa da la tigrnmenki
ga filmo le pu do deukua
@P.i zu nau la tigrnmenki ga takna la .iciron jii du
.i cue du ea noi mutce kejpeo la meicin.
.i de sira zavtedmao .irau tu no kecmao oe tu de
@P.i cue do mi tsodi levi nu turka .ibea mi kasfa ti
.i mi kasfa ta .i klinrmao lemi gasno .i kukra numoi le po mi tcicni
.i noi pristo .i raneu noi pristo .i mi tsodi tau
.i noi mi puntaa pui tu .ipou tu gudbi .e groda helba mi

.ipou mi pa cenja .i mi pana cmiza lepo kerju lo junti
.inurau mi grada pernu ji fregzo
.ipou nana mi danza ba jio mordu tiu
.i mi fomcia lue fizrsensi .e lue ckerao .e lue po pacfurdai
.i nurau ie mi klinrmao lo gasno
@P.i cue du ii tu duvri oe ba jio cmiza nu turka
@P.i cue do ia eu mi kapmao ba jio vedsia je lo ckerao
@P.i cue du loe pacbai ga drudja lo po metflidjimao
.i ei tu drudja da
@P.i cue do ri lea pacbai ga drudja da .ipou mi drudja
@P.i cue du eu tu vedma loe po metflidjimao
@P.i cue do ii .i ii .i sia la .iciron. .i mi fundi tau
.i tau clemao mi roba jio na fatru mi gu ji la meicin. gi neria
.i mi durzo ai letu nu sange .i tau gudbi drudai
.i sia .i tu grada fremi
@P.inau fana le po la sol. ga vreti gu le matfra je loe junti
ga gzotoa da .ice la tigrnmenki ga favgzo ledo hasfa
.i nia le fucmia do takna ledo farfu zea matma le fatru je do
.i cue ledo farfu jii du ia zavlo pua le po tu pa hanblo
la meicin.
.ipou io mi ciktu durzo eu
.i tu takna oe le nu cinta je da le pu da zavtedmao
.i da zavmrotsu na ra loe denli
@P.i cue do ia dui .ice ai dua
.ipou ba gi sui fatru mi
.i noi mi cmiza lemi nu turka
.i neu le po mi drudja lue fizrsensi .e lue po pacfurdai gu gu nukou ie
mi klinrmao lo gasno
.i mi turka ao ba jio mi nardu fomcia ba
.i ii mi fa kapmao ne lea vedsia je loe po metflidjimao
@P.i cue ledo matma ei tu hanco turka ai
.i ai lemi sunho fa nu fomdia pernu
@P.i cue do noi tu prutu pa le po mi cenja ne lea gasklinrmao
.i eo sange ba jio mi durzo kao ba
tie luemi na nu fomcia
@P.i cue du ei tu stise ai le po tu fomcia
@P.i cue do ai noi .i mi fuckri le po mi fia fomcia .e fa kaldru
le po mi nu fomdia
.ipou mi na cnida ba jio nu turka jio gudbi le po kerju loe junti
@P.i cue du ia djipo pua loe po balci loe pernu
@P.i cue do toi bilti .inokou da fatru mi
.i tu jupni ie le po mi metflidjimao
@P.i cue du ie fa petci puo letu patce
@P.i cue do mi fa petci .i mi jurna ba loe po kerju
loe junti
@P.i cue du le metflidjipae ga kusti .i dui pua le vedsia
@P.i cue do ia da kusti .i mi penso ba pei tiu
.i zirbunbo pua le po mi fa petci puo le vedsia .e lemi parti
je levi hasfa gui sui
.i ei tu cusdja
@P.i cue du mi jupni le po io tu djine ba jio spuro metflidjimao
jio gruplilei leba vedsia .ikou le prati je da cmalo io
.icice ii tu cirna pui ba
@P.i cue do mi pa penso ba gi sui .i ␁...␄ toi nardu pui le po cutse ␁.

..␔
.i cue gu do groda brute .i cue do mi nordjilii ao
.i mi danza le vedsia .inurau mi clivi ao vi da
@P.i cue ledo matma ue tei la sunho tu cinta mi ia
@P.i cue do noi .i pia le po letu menki ga nu klogu gu mi valda
.i bleka le korfro je mi .i ei ti cinta
.i noi mi gropeu .icepou noi mi cinta ia .irau mi na nordjilii ao
@P.i cue gu le menki je ledo matma ga dridnpro
@P.i cue du ei tu kapli djifavcea ao letu famji
@P.i cue do ai noi .i mi cnida tu ji la farfu zea matma
.ipou ␁...␔ .ipou mi cnida lo cmalo spasi jio nu ponsu mi
.i ai nifa mi kapli nordjimao mi tu
.i ba jio fremi mi durzo be jia ii gudbi
.i ba ze leba famji kuo titci
.icepou ba sonli vi le parhaa jio nu ponsu ba
.i tu jupni ie be
@P.i cue du mi paziu djano le po na bo tu begco tau
.ipou noi mi fuckri le po bo kukra vetci
.i eo dargzo .inurau mi penso ao .e takna ao letu matma
.i eo srite bu jio lista loe prati je raba jio tu cnida ba io
@P.i cue za nau la moztarci metflidjimao
.i le cmalo kruma ga nu sitfa le teri je lue fordi jua damni loe livsia
.e ga ganta loe vedsia .epou ga nu kusti
.i le skati na draka .ipou le kruma litpae ga nu brili
pui ne lea mahgani tricu pe le nenri hasgardi jia stali ba jio nedza
le balko je le kruma
.i da veslo le ne groda tobme jia snire le frena gu
.e le ne metflidjipae sai ba jio no nu sirna ga jia ia laldo gu
.e le ne taitanholklu jia zarlaldo gi sui gu
.e le ne pedbedpu jia frena le balko .e la tigrnmenki
.i do djomao le metli fespiu jio pazi nu vedma pui do gu
.e le skori ce femsko
.i le bekli jio pendi le darto ga sonpro nana le po ba nenkaa
@P.i cue ba loi .i mi nui kris.
.i eo metflidjimao ti lui mi
@P.i cue do loi .i mi nui tigrnmenki .i ia mi fa dua
.i bia tu neri le fu vedma je mi
@P.i cue la kris. jii du ei .i ti gudbi pui le po satci
.inurau ti sapla
@P.i cue do ta he
@P.i cue gu du berti le ne terjagfro jia gasti gu
jia milmetro tera totnu .e senmetro safera langa via ra le bidje
.e ga nu holdu ra le penta
.ice du berti le ne milmetro nevera kupskotanpae gi sui
.i cue du ti spebi trime loe po favpardji le femsko je le cikrkukpra
datkrilu pe loe merkuri zvobotmotci
.i levi penta ga sitfa le kupskotanpae
@P.i cue do ea rasdjimao da
.i ba ditca pui mi le pu rasdjimao .anoi kanmo
.i ei aa
@P.i cue du ei le zorfotli je lo rasto ga tsufi
@P.i cue do lo saglnrasto ga fotli lo danri gasti lia ti
.i pana lemi fomcirgru mi kutla .e djimao le gasti fespiu
.iza mi bromao da

.i ue lo gasti .enoi lo saglnrasto ga broko
.i nana sui ii le zorhatro gi neu le po metflidjimao fa betmao
le kupskotanpae
@P.i cue du aa ea rasdjimao .i le prati je tiu ga hora
@P.i cue do pengo pitera
@P.i cue du aa
@P.i cue gu do pinkuvmao vi le fa djine sitfa je le kupskotanpae .e le
terjagfro penta tie lo flurnborno flimao
.iza do skodjimao de di
.iza do cabmao ledo torcarterdji linkro .e hatmao le parti
.i do totco de le saglnrasto tirca jia kroli le djisia
.i le parti ga fotli djine
.i do klinrmao da .e favdou da du
@P.i cue la tigrnmenki jia penso gu ui le mi neri turpai .e le mi neri g
e
pengo pitera
@P.i cue za nau la karen. jii du ga nenkaa .e raeble
.i cue du ui le tu ponsia .i mi fundi da
.icepou tu cnida lo durna ce pinti .e lo fordrkla jio gudbi ta
.i ei tu nu helba ao mi le po nurdunmao ti
@P.i cue do ui tu hapci lemi ponsia .i sia .i ia tu gudbi mi
loe po duntia .e lardru
@P.i cue du ei ba vi pa hijra .i cue gu du jugra to ledo hanco
@P.i cue do ia .i mi pa vedma pui neba .i mi pa jurna le pengo pitera

@P.i cue du ui .i cue gu du barnraegzo do .e skesa do
.izi do skesa du bai
.i le pu ponsia ga secfurpoa pui do ze du
.ikou ba rodja vi le dampai
.i de [set referent] siltu lede korti .e skesa
.i la tigrnmenki ga favdjimao ledu lapla .inia la karen. dui
ledo rirgnkua
.izi de felda le pedbedpu
@P.iza le darto bekli ga sonpro .icice nenkaa pua ba jio zirlaldo
fumna jia berti le kanra zea datkrilu
.i leba menki ga kubcea
.i ba satci le po ba santi zvogzo
@P.i letau nu bivdu ga kusmo
.i le matfra je do kuo dua .ice do kuo dua sui neu loe po do canhia da
.ipou le fu bivdu ga norkusmo .imoi cue do eo stolo .enoi dargzo
.i cue gu do zvogzo le pedbedpu ji trati [ji is x.conn]
.ipou la karen. gi sui dui .ikou do .e du genrnenfelda duo ba jio
nanda lue barma zea tugle
.i cue do ea minpismi .i duo ie mu selrezmao
.i stolo fia le po mi zvoto
.i cue gu do rescle taicea le fordi .e ga fu kenti
puo le fumna pui be gi tou
.i tu nu helba eo mi ie
@P.i cue le fumna jii di tu jia deknie ga briga pui le po solcou
.i cue di jia clafo gu eu tu nurkuvmao letu pingu
@P.i cue gu do resdji ledo rirgnkua .ice du resplicea .e skitu
le fordrkla jia snire le pedbedpu
.i cue do uu mi fatru tu .ice uu mi pa nordaspa tu gi eu
@P.i cue di tau cmalo fatru .ipou ae noi tu pa kokfa ba jio cinta

@p.i cue do ia noi mi .e ta zavmrotsu favdaspa lia tiu le cinta gi eu
je mi ze ta
.i mi .e ta djano le pu mi .e ta deknie
.i mi ze ta pa sira zvokitsa
@P.i cue di ui .i hapci .inau ei tu nakso kao levi kanra
@P.i cue gu da nu karku vi be jio bamtanjuo jia zarfrezi trana
.i cue do ti zvopendi .i ei ba pa bloda ti
.iconoi ei ti pa broko nia le po norma nu bapra
.i mi vizka lo rajmra le kanra gi va le tanjuo
.i ii da ripa naksnsifsea .i ei
@P.i cue di eo slano .i tu nu ge citlu menki
@P.i cue do jia cusbromao gu mi nui tigrnmenki
@P.i cue di toi gudbi
.i ti norma nu bapra pa le po ti broko
.ice tu drecue pei le tanjuo
@P.i cue do mi kanmo le po mi metflidjimao le kanra
.i da fa clina io
.ipou io tau nu gudbi .inurau ii da .e le tanjuo ga fibru
pui le nu matci je da
.i mi sange ba gi tou
.i ea sifsea le kanra jio cronikfernu .e ne lea rolgu tanjuo
.ipou uu noi mi balci kao letau parti
.inurau lemi patce ga purcu .e sasurura
.ibea noi mi kutla kao le kurfa holdu jio sitfa le tocki
jia djimao le datkrilu le kanra
.i ea tu godzo le macdru vedsia jio gudbi ti .i uu
@P.i cue di sia uu .i mi petci oe le hora tu
@P.i cue do tu petci ou .inurau mi niba pa durzo
@P.i cue di ia tu groda helba mi
.i tu spuro ce faspeo .inonukou tu sekci
.irau ae tu fa durzo le nu turka
.i ie pa ditca tu .iji duo ie tu spuro cenja
@P.i cue do le cirgru nujie mi je lue po pacfurdai pa fomcia
vi la gruman. kompi
.i ei tu fu cninu da
.i le pacfurdai jiu da pa ditca pui mi .e pa donsu pui mi
lede nu spuro jia na nu gendou mi tu
@P.i cue di numoi ie dui
@P.i cue do ue tu djano noi
.i le cteki ga ju petci oi lo po ditca
@P.i cue di niba pa cutse toi mi
.i ei dua ii puo lemi kompi .i sia tigrnmenki
.i tu groda helba .i loa
@P.i cue la karen. jii du mi solcou ia pa le po le laldo fumna
ga nenkaa
.i numoi ie tu nakso noi leda bekti
@P.i cue do noi da laldo [comment that in-mind ref doesn't fit pred]
.i nu fundi mordu pui mi pua le po mi takna da le gudbi nabretpi gu
pue le po mi tokna leda cmeni .e zavlo nakso
@P.i cue du ia tu rasto pa le po tu frena da
@P.i cue do liu rasto sanpa ie tu
@P.i cue du de sanpa le pu no comtu ze comtu oe
.i de trida gleca
.i tu solcou oe .inumoi tu jia rescle vi stali .e lia takna da

@P.i cue do nurau ie noi dui .i noi mi pa zavdru
@P.i cue du noi tau kusmo
@P.i cue do tau he kusmo
.i raba rescle sucmi .irau nurau ie noi ba rescle vedma oi
.i le zvoto ga brili
.i ea kalmao ba jio nu satci
.iza ea banci .e titci le midmia
.i na lena genza mi srite be le darto .e sluklo de
@P.inau fafana le midmia do .e du favgzo le vedsia
.iza nenkaa pua la .iciron. jia cutse ba gi tou
.i loi .i mi vi kamla .inumoi mi bleka ao letu ponsia
.i ti gudbi .i tu ponsu lo patce
.i eo mi pleci ti
.i cue gu la .iciron. tokna ne lea groda skori ce femsko
@P.i zi fadgzo pua ba jio fu vedma jia pucto be jio
hankarti jia le frama je be broko
.i cue ba ei tu nakso kao ti
@P.i cue do ia .i tau sapla .i vi rolgzomao
@P.i cue ba fa ie tu durzo ai
@P.i cue do le nadzo .i ␂...␃ .i le prati ga pengo pitera .i ei aa
@P.i cue ba aa
@P.i cue la .iciron. eo mi katca
@P.i cue do noi .inurau le zurblanu litla eu cabmao tu
.ipou tu katca oi le po mi bremao le djisia
.i cue gu do klinrmao ce pilmao le metli tubli jia pa nu karkncea
vi le djisia gu tie le motci ge rofcti krilu jia nu hanbapra
.i rordirkro pua le klada nujie lo fagro ge fernu cmacai
@P.iza cue do eo raba zvogzo
.i cue gu do djimao le dertu tirca le frama
.i do sacbapra le metflidjipae
.i do resplicea ledo gluva ce frekosta ce rilcurkapma
.i do sacmao le lenki borku .e zi kukra djimao le parti jio pa broko
.i cue do eo genrnenkaa .i mi pa kaldru
@P.i cue le fu vedma ti gudbi ia ti gi pa
.i cue gu da petci .e zvogzo
@P.i cue la .iciron. jii di ia mi fundi lue fagro cmacai
.i lo litla via le darto damhou ga gutra
.i tu cnida ba jio sitfa jio lo fu vedma ga pazda oi vi ba
.i pa purcu pua le po tu fitsui mia le dzosia
@P.i cue do ia .i ii mi cnida ne lea codji [␂screen, from Japanese␃]
.i ei mi fu vedma conoi balci da ao
@P.i cue di jia clafo ce cusbromao gu eu tu rojmao da
@P.i cue gu do nurnitmao lue herfa je di jia kovgzomao ledi hedto
.i cue do sia le ditca .i mi cutse oe li duo ie mi getsi ao ba jio
codji lu
.i noi mi fu vedma ao da .inurau loe batmi codji ga nu litrtce
.icimoi mi pinkuvmao oa lede bilti durna
.i mi fa balci da
@P.i cue di .ice tu cnida ba jio cersi
@P.i cue do mi skitu le fordi
.inoca oi le fu vedma gi sui jue mi skitu de
.i lua skitu je loe fordi ga nubie tu jia ponji
.inosoa nukou ie tu fulrndru
.ipou ai mi fa fu vedma le fordrkla gi sui ␂...␃ .i dui fa le futnerdei

@P.i cue di mi fa hapci le po mi kinci tu tiu
.ipou io la malin. na blipeo le po mi kamla noi
.irau mi favgzo oe le grupa .i loa
@P.i cue do .e du loa la .iciron.
@P.i cue la karen. mi gi sui hijra oe lemi fomcirgru .i loa
@P.i za nau do skitu le fordi .e fomcia lua numsensi
.i sei [spebi] do srite ba jio prupre
.i le darto bekli ga sonpro .ice ba jio fu vedma ga nengzo
.i cue ba mi cnida feni be gi lia ti na ra lea likta
.i ei tu balci kao be
@P.i cue do ui ia .i cue gu rebo jio metflidjimao vedsia ga jurna
piro lo cmeni bu jio rilri nu turka lia tau
@P.i cue ba tu jia deknie ga junti .ice tu cninu pui mi
.i eo pruci metflidjimao nebe gio jia nu bleka mi
@P.i cue do ia aa .ipou mi ponsu sine le frekosta
@P.i cue ba mi nigro .ikou ne le po mi fu litla lo zurblanu
fa fu surna mi noi
.i ui tu ponsu to lea rilcurkapma
@P.i cue do jia penso .enoi cutse gu tu daspa letu skapi
.i cue gu do djupo le parti .e djimao le tirca .e metflidjimao le nubiu

.ineu le fu vedma ga citlu katca
@P.i cue ba le drufro pe tu ga tsufi
.i cue gu do penso le pu da gudbi le sira tsufi
.i cue ba eo za prudru pui le djine pue le po nartrcpu de
@P.i cue do ueuu .i noi mi ponsu le nartrcpu prupae
@P.i cue ba ei ia .i uu mi rulbeo le po rabe jio kuona turka lemi nubiu
pazi prudru pui be
.i eu be malveo .a fleti puo lo ganja .irau da delfrezi oe
@P.i cue do ii noi mi turka kao dia tu
.ipou eo srite letu namci ce numcu
.i mi fa sritaa tu fa le po mi getsi ne lea nartrcpupae
@P.i cue ba tiu gudbi vidre .i eo mi sange be tu
@P.i cue do aa
@P.i cue ba tu cnida le nartrcpu prupae .e le metli tobme jio trana
.i de .e da djipo patce
.i ea djine bo jio spuro metflidjimao .inumoi letu vedma prati
ga cmalo da gi na .icice tu cirna pui bo
.i curmao letu djoto .inurau nabu le drida je lo metli flidu
ga rolfelda di le tobme .i tiu pundou
@P.i cue do sia .i ia tau gudbi nu sange
.ipou lemi frekosta ga kuvga lemi fitpi
@P.i cue ba rena dui .i mi petci oa hora tu .i lo pengo pitera ei
@P.i cue do lo nira .i tu petci le pu spuro
@P.i cue ba noi .i mi plizo letu livcke ce ctifu
.isoa mi petci oa .i tokna ti .i loa
@P.inau nana le po la sol. ga vrecea gu la karen. jii du ga favkaa
.i cue du ea hapckedru pei letu vedsia
.i ho le fu vedma ga hijra pia le po mi nu notsia
@P.i cue do ne le fu vedma .i da tifru lo rilri nu turka
.ipou da no nu tsufi io mi
.i uu .ipou mi jurna le pengo pivera nia le nazdei
.irau mi hapci

.i mi penso le po eu mi ze tu fa kokfa le fucmia
.inumoi di pluci lemi matfra .icinukou di vizmao de le pu
noi mi tokna lomi cmeni .e za brogzo .i ei tu togri
@P.i cue du ia .i ea mue hapckedru
.i mu kokfa ie .i ei tu jia mendi pa penso tiu
@P.i cue do ia mi pa penso tau
.i mue fa titci lo nidlrhadskapi gi duo la neubyrg. ga
.e lo groclife brasica
@P.i cue du ei .i da favhamcmi .i da nu tasgu
@P.i cue do da gudbi nu gusto ia
.i ei tu pa gutsae pui lo brasica gi duo le pu jungi
r.i eo curkri mi da
@P.i cue du aa .i ea za fu vedma lo tcidi
.i ei tu klipu lotu cmeni .i le nidlrhadskapi fa kusti
@P.i cue do de nu sitfa lemi beldndjale
@P.i za nau do ze du nenbei le sakli je lo tcidi ledo hasfa
.i cue do loi matma .i cue du loi dzou taitai
.i loi cue le matma jii di
.i cue do ie sitfa la farfu
@P.i cue di ba jio matci ga broko papazi da kamla ao
.i da fazi kamla .i cue gu di hapnerjnfoa noi
@P.i cue do ui bleka lemu fucmia
.i cue gu do lufta puo le sakli le to nidlrhadskapi
jia fibru siltu leda tugle
@P.i cue di ue .i tu cmepli da oe noi
.i tu favsromao oe noi
.i tu cnida lo cmeni le po fomcia
.i tu dargzo letu matma .izi tu brize bivdu
@P.i cue do jia kerju troli ledo deknie kadsmina gu noi mi pa favsromao

.i mi cao jurna ia nia lena denli lo cmeni jio ju vedma
pue ti
.i ae mu hapckedru pei le pu mi gropeu bivdu ji kanmo le po dui
@P.i cue di noi .i tu rafa cinta mi .i cue gu di satci
le po dridnkra
@P.i cue do uo tei matma .i noi mi cinta .i mi pia valda
.i ea hapci letu valda cinta
@P.i cue di ei ea .i ea he .i eu mi durzo noi le nu sange je tu
.imoi ei tu dargzo mi
@P.i noi tei la matma .i raba clivi traci .icu ba fundi tiu
.i eu tu stimuo .inomoi mi kanmo noi le po mi gi sui stimuo
.icinurau mi .enoi tu biu le prase je le po mi clivi
.i ii mi nu kliri takna ␁...␂
@P.i cue du jia cusbromao gu ue le hadskapi ga selrezmao
.i cue ia da slano dargzo le monca je lo tcidi
.i du nu fasru le po eu du litnu da
.ipou du cusbromao ai
@P.i do gensea da le monca
.i cue do jia dreti ponda pui du gu tei la matma
eo sange ba mi
.i mi pa fu vedma lo brasica
.ipou ii noi da harmo lo nidlrhadskapi gi duo la neubyrg.
.i mi kokfa oe ie jio liftcidi
@P.i cue di .inorau tu cnida letu matma

```
@P.i cue do ia  .i mi rafa cnida ce cluva tu
.inonurau mi valda
@P.i cue di jia cmalo crano gu ea kokfa lo baitsai ze lo citrnhebkapma
.i da harmo mordu lo brasica lo hadskapi
.i lo brasica ga harmo lo kasnrmio ca pojmio
@P.inau le farfu je do ga nenkaa
.i cue da jii bo ue he pua ta jio dzoru puo le tobme
.i cue gu du genza litnu le hadskapi
@P.i cue do mu hapckedru pei lemi vedsia
.i mi kambei lo nidlrhadskapi lemi famji
@P.i cue bo mi vizka  .i ei tu fia djupo le famji
@P.i cue do ueuu  .i ue mi fatcenmao tu gi sui
@P.i cue bo jia clafo gu tu djupo oi .iconoi tu gropeu durzo
fapa le po tu bredi
@P.i cue do jia damslomao ledo menki gu mi drucue le pu mi rispe
ce prigzo tu
@P.i cue bo jia nurnitmao luedo herfa gu noi tu durzo oa ro le sanpa
je le pu selnurporsea
.i tu gudndru pou oa
.i tu gudbi junti kii deknie  .ikou ia fa dui
.i mi rispe tu le po tu daspa tu  .i ea hapckedru
.i tei la matma ei tu lia filmo
@P.i cue di noi mi fundi le pu do nordjilii
.ipou ia do gudbi deknie .ice ii mi fuckri do oa
@P.i cue bo toi fekto
>@ <EOF
```

metflix.txt

```
>@XMIT DX1:METFLIX.TXT
@H@RLa Metflidjimao Vemsia@XTranslation / Carter@X1982.1.11@W1,
@F@W1,@C@N
@B@CLa Metflidjimao Vemsia
(The Welding Shop)
James F. Carter
@W1,@E
@PTigereye watched the waves from the kids in the canal.
Their high voices grated on his ears.
The horizontal rays of the just-risen sun illuminated a small
fight in the water.
Mei Chi called, "Tigereye, Tigereye!  He dunked me!
Punish him!"
@P"Punish him yourself," Tigereye said.  "Nag at him, or
dunk him back.  Don't always ask me to straighten out your problems."
Mei Chi was not much pleased with his help.
@PHe brushed a black hair from his slanted eye, and thought,
"What's making me sad?
i'm doing well in school.
My teacher just complimented me on my history paper."
He stroked his strong muscles and his handsome mahogany skin,
which usually pleased him, but not now.
"Suppose I take a swim," he thought.
"Ichiro!  Let's race to the next building!"
```

He took off his loincloth and tossed it in a box on the float, and jumped
into the water. Ichiro was a strong swimmer, but not so strong
as a teenager. Even so, the race was enjoyable.
@P"The barge!" Tigereye cried. "Swim over here right now!
Joseph, move your ass! Do you want to get chewed up by the barge's propeller?"
He thought, "Damn, I have to watch them every minute.
Why can't they take care of themselves?
I swim for one second and they get into trouble."
@P"I have to pee."
@P"Go, but be quick about it." The boy pulled himself out of the water.
@PThe rising sun reminded Tigereye that he was hungry.
"What's taking that kid so long to pee?"
he thought.
Finally he returned.
"Lunchtime!" Tigereye called. "Everybody get out and dress.
Follow me. Gloria, your laplap is coming off.
Come on! Move it! Don't lag back!"
He dragged his charges to his house, which was on the fifth floor
of one of the big apartment buildings.
He heated the food in a microwave oven.
One of the little ones asked, "I'm hungry. Please hurry."
@P"I can't cook any faster. Please be patient."
In a while he fed the children.
@P"Mei Chi", he shouted, "that's the most disgusting thing
I ever saw! Go to the corner! Never mess up your food like that!"
@PMei Chi went to the corner and cried. "I'm hungry.
I'm so hungry."
@P"You should have remembered before you smeared up your food.
Now shut up." She continued to sob.
@PEventually all the children finished eating.
Tigereye, though, was still hungry, because the food was not quite
enough for everybody and he had eaten a small serving.
He took his backpack and said, "The group is going to the park.
The older children will study."
@P"Aaaw!"
@PIgnoring that, he went on, "Get your packs, and let's go."
@PRicky, who was four, announced, "Caca."
@P"Damn. Ichiro, please take charge. We'll meet you there.
Go and do it, Ricky. I'll wait."
@PHe crapped, then said, "Wipe my ass."
@P"Wipe your own ass. You know how."
@P"So why are you waiting for me?"
@P"Wipe. Do you think you can be left unsupervised?"
@P"I'm very responsible."
@P"Right, you're more responsible than most four-year-olds.
But you can't be left by yourself.
Stick up your butt. Clean enough. Wash your hands."
@PWhen they rejoined the group, Ichiro was bearing up strongly but was
near tears, for Mei Chi had been refusing to obey him.
"Mei Chi took Mzima's car and refuses to give it back."
@P"Mzima gave it to me," Mei Chi said.
@PBut Mzima said, "No. I just shared my car with her.
She's bad and nasty by stealing something shared."
@PTigereye judged, "Right. Give the car to Mzima."

@P"No! You're playing favorites. He gave it to me,
and I won't give it back!"
@PTigereye sighed and said, "Please don't make trouble. Be good."
He felt that Mei Chi was trapping him into some kind of mistake.
@PShe shouted, "Favorites! Favorites! Favorites! I hate you!"
And she hit Tigereye with the car, which cut his arm.
Without thinking he slapped her face, to which she cried,
"You're bad! You wounded me! You're irresponsible!" Etc., etc.
@PThe subsequent punishment was loud and long, and afterward
Tigereye felt filthy.
@PLater, Tigereye was talking with Ichiro, who was saying,
"Don't take Mei Chi so seriously. She's just a brat, so
you shouldn't get all sad over her."
@P"I hate this job. I punish this one. I punish that one.
Wipe my ass. Quick, I'm hungry. Don't lag back.
rAlways don't lag back. I hate it!
I'm not complaining about you; you're a good one and a big help to me.
But I've changed. Once I enjoyed taking care of kids
because I was a big man, a leader. But now I want something more.
I study physics and calculus and engineering.
Why should I be wiping asses?"
@P"Maybe you should find some enjoyable work."
@P"Yeah. Maybe I could open a calculus shop."
@PIchiro replied, "Engineers know how to weld. Do you know
how to do that?"
@P"Some engineers know that. I do."
@P"How about offering a welding service?"
@P"Maybe. Hmmm. Thanks, Ichiro, I like that.
That would get rid of several of my problems: Mei Chi, number one.
I'll certainly take your suggestion. That's a good idea.
Thanks! You're a good friend."
@PWhen the sun rose to vertical, the parents of the children
collected them, and Tigereye returned home.
During dinner he talked with his father and mother about his troubles.
His father replied, "Certainly it was wrong to slap Mei Chi, but I think
I would have done the same thing.
You should talk to her parents about her brat behavior.
She's excessive, every day."
@P"She certainly is. I'll do that. But there's something else
bothering me. I don't enjoy my work.
I know physics and engineering, so why am I wiping asses?
I want to work at the things I'm studying so hard.
Perhaps I will open a welding shop."
@PHis mother interjected, "You're going to do manual labor?
My son is going to be an educated person!"
@P"You didn't complain when I became an asswiper.
Suggest something else I can do with my education now."
@PHis father asked, "Do you intend to stop your education?"
@P"Certainly not. I expect to stay in school and finish my education.
But I need work now that's better than taking care of kids."
@P"Building people is an important job."
@P"Those are pretty words. But the job bugs me. What do you think
about welding?"
@P"Who's going to pay for your equipment?"

@P"I'll pay. I do earn something from babysitting."
@P"Welders are expensive. So is a shop."
@P"Yes, they're expensive. I had an idea about that.
It's a little dumb for me to pay for the shop, plus my share of this
house. You understand?"
@P"I think probably you should join up with an experienced
welder who would share his shop, so your expenses wouldn't
be so much, and you maybe could learn something from him."
@P"I thought this way ... This is hard to say." Tigereye took a
deep breath. "I want to move out. I want the shop so I could
live in it."
@PHis mother said, "Son! You're my baby!"
@P"No. While your eyes were closed I grew up. Look at my body shape.
Is this a baby? I'm not an adult, but I certainly am not a baby.
That's why I want to move out."
@PTears came from the mother's eyes.
@PHis father asked, "Do you want to completely break with your family?"
@P"I don't intend that. I need you, Mom and Dad.
But ... but I need a little space of my own.
I never intend to completely split from you.
One of my friends has an arrangement that might be good.
He and his family usually eat together, but he sleeps in his own
apartment. What do you think of that?"
@P"I've known for a long time that you would ask this.
But I didn't expect it so soon.
Go away, because I want to think and talk with your mother.
And write down a list of how much the stuff you need will cost."
@PLater: Morning Star Welding.
The small room was on the third floor, which was low for a residence
but high for a shop, but cheap.
The sky was dark, but the room's lights illuminated a mahogany tree
growing in the courtyard that rose just beyond the balcony of the room.
Inside was a big table near the front,
and a welding generator of uncertain origins, but which was
definitely old, a drill press, also rather old,
a hammock in front of the balcony -- and Tigereye.
He was sorting metal scraps which he had just bought, and nuts and bolts.
The bell hanging from the door rang as someone entered.
@P"Hi. I'm Kris. Would you weld this for me?"
@P"Hi. I'm Tigereye. Sure, I'll do it. By the way, you're my
first customer."
@P"Really? This is good for a beginning, because it's simple."
@P"What is it?"
@PKris was carrying a triangle of steel 3 mm thick and 5 cm on
each edge, with a hole in each corner. He also had a
19 mm wrench socket.
"This is a special tool for taking the timing gear nut off a
Mercury outboard motor. This point is where the socket goes."
@P"Let's braze it. I was taught to braze when you can. OK?"
@P"Will the brass be strong enough?"
@P"Silver-brass is stronger than ordinary steel like this.
In my class I cut and rejoined a steel scrap, then broke it.
The steel, not the braze, broke.
Also, maybe the heat of welding will distort the socket."

@P"OK, let's braze it. How much will this cost?"
@P"0.3 pengo."
@P"OK."
@PTigereye painted the joint surfaces of the socket and the
triangle point with fluoborate flux, then bolted them together.
Then he lit his acetylene torch and heated the parts.
He touched them with a silver-brass wire which flowed into the joint.
The parts were strongly joined.
He cleaned up the assembly and gave it back to Kris.
@PTigereye thought, "My first job -- and my first 0.3 pengo!"
@PAfter a while, Karen came in and looked around.
"Great, your own place! I like it. But you need some decorations and paint,
and a better rug than that. Would you like me to help you
fix this place up?"
@P"I'm glad you like my place. Thanks, you're much better than me
at decorating, and art generally."
@P"Has anyone been here?" She took his hands in hers.
@P"Yes. I had one customer. I earned 0.3 pengos."
@P"Great!" She threw her arms around him and kissed him.
He kissed back.
Their privacy excited them, and something grew down below.
They wriggled their bodies and kissed.
Tigereye loosened her laplap, while Karen took off his loincloth.
They fell into the hammock.
@PWhereupon the doorbell rang, and in came a middleaged lady
carrying a shaft and gear.
Her eyes widened, and she began to silently leave.
@PThis behavior was customary. Tigereye's parents would have done
the same, and he would have too if he had come upon them by chance.
But the situation was not customary. "Please stay! Don't go!"
He got out of the hammock -- that is, he tried.
But Karen also was trying to get out, and both of them flopped back
in a tangle of arms and legs.
"Let's keep calm. How are we going to get out of this?
Hold still until I'm out."
He nakedly stood on the floor and asked the woman,
"How may I help you?"
@P"Teenager, you're a bold one in an embarrasing situation."
She laughed. "You could cover your cock."
@PHe put on his loincloth, while Karen dressed and sat on the rug
near the hammock.
"I'm sorry to trouble you, and sorry to forget my responsibilities."
[to be irresponsible to the hypothetical you, i.e. before you became
actual by entering the shop.]
@P"No big deal. But I hope you weren't cooking up a baby."
@P"Of course we aren't that irresponsible [to our hypothetical future
child]. We know we're teenagers. We were just making out."
@P"Good. Have a good time. Now, can you fix this shaft?"
@PIt was cracked at a ball bearing, which turned fairly freely.
"This shaft overhangs. Did something hit it, or did it break in
normal service? I see some scratch marks on the shaft near
the bearing. Maybe it's been replaced several times. Right?"
@P"Slow down. You have sharp eyes."
@PHe interrupted, "I'm Tigereye."

@P"That's appropriate. It was running normally when it broke,
and you're right about the bearing."
@P"I can weld the shaft, and it would probably even come out straight.
But probably that's not such a good idea.
Perhaps it and the bearing aren't strong enough for their job.
I have a suggestion. Substitute a stainless steel shaft,
and a roller bearing.
But unfortunately I can't make those parts, because
my equipment is poor and insufficient.
For example, I can't cut the square hole for the key that holds
the gear on the shaft.
I suggest you go to a machine shop better than this. Sorry."
@P"Thank you. How much do I owe you?"
@P"No charge. I didn't do anything."
@P"Sure, you helped me a lot. You're expert and smart, even if sexy,
so I had hoped you would do the job.
Who taught you, that is, how did you get your expertise?"
@P"My engineering class studied at Grumman Company.
Do you know them? Their engineers taught us and gave us
their experience, which I pass on to you.
@P"What got them to do that?"
@P"Don't you know? They can pay taxes by teaching."
@P"Nobody told me that. Could I maybe do that in my company?
Thanks, Tigereye, you've been a big help. Bye."
@PKaren commented, "I was so embarrassed when the old lady
walked in! Why didn't you fix her thing?"
@P"She wasn't old. I'd rather talk to her about a good solution
than take her money and mess it up."
@P"You were certainly brazen in front of her."
@P"'Brazen' means what to you?"
@P"It's English slang. It means unashamed when you should be.
You should have been embarrassed to stand
there naked and talk to her like that.
@P"Why shouldn't I? I didn't do anything wrong."
@P"It's not customary."
@P"What customary? Everybody swims naked, so why not sell naked?
It's light outside. Let's finish what we started, then
bathe and get some lunch.
And this time I'll put a note on the door and lock it."
@PAfter lunch Tigereye and Karen returned to the shop.
Soon after, in came Ichiro, who said,
"Hi. I came over to see your place. It's nice. You have equipment.
May I play with this?" He took a big bolt and nut.
@PRight then, a customer arrived pushing a handcart whose frame
was broken. "Can you fix this?"
@P"Sure, that's simple. Roll it over here."
@P"When will you do it?"
@P"Right now. The ... The price is 0.3 pengo. OK?"
@P"OK."
@P"May I watch?" asked Ichiro.
@P"No, because the ultraviolet light would burn you up.
But you may watch me prepare the joint."
He cleaned and levelled the metal tube, which had cracked at a joint,
with a motorized hand-held abrasive wheel.

A cloud of iron sparks spewed out.
@PThen he said, "Everybody please get out."
He connected the earth cable and turned on the welder.
He put on his gloves and apron and flare shield.
He struck the electric arc and quickly joined the broken parts.
"Come on in! I'm done."
@PSaid the customer, "This is sure better than before."
He paid, and left.
@PIchiro commented, "I liked the burning sparks. And the light under
the door was wierd. You need someplace for customers to wait.
It's crummy to kick us out in the hall."
@P"Right. Maybe I need a screen. But do I want to buy or build it?"
@PIchiro laughed and interrupted, "Maybe you could grow it!"
[Tease for not asking free question.]
@PTigereye mussed his hair as he ducked his head. "Thanks, teacher.
What I should have said is, 'In what manner do I want to acquire
a screen?' I don't want to buy it because commercial screens are
translucent, so I would have to paint over the pretty designs.
I'll build it."
@P"And you need a chair."
@P"I sit on the floor, so my customers can sit there too.
You're a floor-sitter, Jap, so what makes you so fancy?
But I certainly will buy another rug -- tomorrow."
@P"I'd like to go with you on that. But Maleen is probably
thinking maybe I'm not coming back. I'd better get back to the group.
Bye."
@P"Bye, Ichiro," Tigereye and Karen said.
@PKaren added, "I have to get to a class too. Bye."
@PSome time later, Tigereye was sitting on the floor doing math.
Specifically, he was writing an examination.
The doorbell rang and a customer came in and said,
"I need fifty of these per week. Can you make them?"
@P"Wow, sure!" A welding shop earns much of its money
from regular work like that.
@PThe customer replied, "Teenager, you're young and unfamiliar
to me. Would you make one while I watch, as a test?"
@P"OK. But I have only one apron."
@P"I'm black, so one exposure to ultraviolet won't hurt me.
I'm glad you have two flare shields."
@PTigereye thought, but didn't say, "It's your skin."
He supported the parts and attached the cable and
welded the assembly, while the customer watched closely.
@P"Your technique is adequate." Tigereye thought his technique
was more than just adequate. "Now would you proof-test it by
pulling it apart?"
@P"Oops! I don't have a pull-tester."
@P"Really? Unfortunately, I have a rule that anyone who works
on my stuff has to test himself immediately before.
If he's sick or flying on pot, he should take a day off."
@P"I guess I can't work for you. But would you write down your
name and number? I'll send you a message when I get a pull-tester."
@P"That's a good idea. May I make a suggestion?"
@P"OK."
@P"You need a pull-tester and a rotating metal table.

Those are important equipment.
Why don't you join an experienced welder, so your overhead cost will
be less and you can learn from him?
And protect your toes. Sometime a drop of molten metal
is going to roll off the table on them. That hurts."
@P"Thanks. Those are good suggestions.
But my apron covers my feet."
@P"Usually. How much do I owe you? 0.3 pengos?"
@P"Nothing. You paid with experience."
@P"No, I used your time and materials, so I should pay.
Take this. Bye."
@PWhen the sun got vertical Karen came back.
"Let's celebrate about your shop! How many customers were here
while I was gone?"
@P"One customer. He offered regular work, but he wasn't satisfied
with me.
["io" is QPI, not per L1. He believed probably I wasn't adequate for him.]
Unfortunately. But I earned 0.9 pengos today, so I'm happy.
I think, let's cook dinner, because that would please my parents,
and show them that I'm not going to take my money and split.
Do you agree?"
@P"Yes, let's celebrate with your parents. What shall we cook?
Male, did you think about that?"
@P"Of course I thought about that.
We'll have lobster Newberg, and cabbage."
[Lobster cabbage style?? Does "ga" get the conjunction joining
the right things?]
@P"What? That's a joke. Cabbage is disgusting."
@P"It tastes good! Have you ever tasted cabbage Chinese style?
Trust me on this."
@P"OK. Let's buy the food now. Do you have your money?
The lobsters will be expensive."
@P"Right in my belt."
@PSoon Tigereye and Karen carried the bag of food to his house.
"Hi, mother!" "Hi, Mrs. Chow!" [Translate in Chinese, which turns
out here to be lexable Loglan: loi dzou taitai]
"Hello," said his mother. "Where's Dad," Tigereye asked.
@P"Some machine broke just before he wanted to come home.
He'll be along soon." She was not full of enthusiasm.
@P"Look what's for dinner!" Tigereye lifted up the two spiny
lobsters from the sack. They feebly waved their legs.
@PThe mother said, "Hey! You shouldn't have spent for that!
You shouldn't dig into your savings. You need the money
for your education.
Leave your mother, and you run wild like the wind."
@PTigereye replied, carefully controlling his teenage temper,
"I didn't take from my savings. I earned today the money I used
to pay for these. I hoped we would celebrate that I'm
adult enough to do that."
@P"No, you'll always be my baby!" She started to cry.
@P"Mother! I'm not a baby! I grew up. Enjoy your grown-up baby."
@P"What's this? Are you telling me what to do?
Suppose I don't, are you going to walk out on me?"
@P"No, mother. Everybody goes through life whether they like it or not.

```
If you stop, I can't also, because it's me, not you,
that's living my life.  Maybe that isn't very clear..."
@PKaren interrupted, "The lobsters escaped!"
They were actually slowly leaving the pile of food,
but she could easily have restrained them.
But she intended to interrupt.
@PTigereye put them back on the pile.
Correctly taking his cue from Karen, he asked,
"Mother, would you give me some advice?
I bought cabbage.  But maybe it doesn't go so well with lobster Newberg.
What vegetable should we cook?"
@P"So you still need your mother."
[Moving out, even so (.unorau) you need me.]
@P"Yes, I'll always need and love you even though I'm grown up."
@PShe gave a small smile and said, "Cook bai tsai and mushrooms.
That goes better than cabbage with lobster.
[da = "that" = the set, both foods -- Nalgol augmentation.]
Cabbage goes well with beef or pork."
@PTigereye's father arrived.  "Hey, what's that walking on the table?"
Karen again restrained the lobsters.
@P"We're celebrating my shop.  I brought spiny lobster for my family."
@P"So I see.  Are you going to support the family now?"
@P"Oh, crap.  I've upset you too."
@PThe father laughed.  "Go ahead and support.
But don't make like an adult before you're ready."
@PTigereye lowered his eyes and replied, "I promise to
respect and follow you."
@PHis father mussed his hair and said, "You don't need
to make all these signs of submission.  Just do good.
You're a good kid, I mean teenager, so you will, I'm sure.
I respect you for taking responsibility for yourself.
Let's celebrate.  Mother, is that how you feel?"
@P"I don't much like that he's moved out.
But he is a good teenager, and I guess I have to trust him."
@P"That's a fact."
@W2,@CGeneral Comments@W2,@E
@PThis story was written "directly" in Loglan, in the sense that I
did not first make an English draft of it.
It is not true that I can think in Loglan (yet), but I tried
to simulate the result that might have been had I in fact
composed the entire text with no reference to English at all.
Thus, I treated English slang, literary conventions, etc. as
out of bounds.
In particular, I did not use the pervasive past tense that
English stories have.
@PThis English translation is rather literal, so that style problems
in the Loglan can be recognized more easily.
@PI feel like my vocabulary is at about the fifth grade level.
Now this is far better than I can do after an equivalent amount
of study of French and Russian -- in those languages I am barely
competent to handle the minimum subset grammar, much less
a decent vocabulary.
Nonetheless I feel frustrated.
Many times I translated an internally coded meaning into Loglan,
```

only to find later that the Loglan word I used was a whole lot less
specific than the English one I would have chosen.
On the other hand, sometimes I could be more specific in Loglan.
We shall have to see what, if anything, ought to be done about
my Loglan vocabulary.
Perhaps I should go over the story and clean up
words that are not as specific as they might be.
@PEnglish has a rich set of literary conventions, which of course are
absent in Loglan.
I particularly missed "he said", "he cried", "he murmured", etc.:
the side channel for feeding in information about the manner and
expression of the speaker.
You will notice that I achieved their effect somewhat by, for example,
"cue do jia clafo" etc.
These expressions seem clumsy to me, but maybe this is just because
they aren't the English I am familiar with.
@PJudged aesthetically, my "cue do" convention worked out adequately,
though it isn't as agile as the English quoted strings.
However, it is syntactically and semantically unambiguous.
As anciently defined, li-lu quoted strings are unitary and opaque to the
parser.
If this be accepted, the narrative and the strings are parseable
(separately), but it is impossible to coordinate pronouns, tense reference,
etc. between them.
A later proposal (formally implemented in the T.19 MacGram)
was to parse quoted strings with the narrative and to ignore
the quotes when doing pronouns etc.
Unfortunately, I fear that when the Understander goes to work
on multi-sentence quotes, the parser will have not yet completed
parsing the embracing narrative, leading to a stack corruption or worse.
Also, one traditional use of quoted strings is for exact quotes;
English of course is not super clear about when the quote is exact
======= BROKEN HERE
and when pronouns etc. are to be coordinated, but it doesn't matte
particularly
if an artificial intelligence is going to process this stuff,
so I identified the speaker in every sentence, one way or another.
@PI had trouble controlling forcefulness.
The sequence where Ichiro suggests the welding idea is a good example,
where a crescendo effect is desired but not perfectly achieved.
"cao" here would be overkill.
I find myself using "ia" for emphasis: "mi solcou ia pa le po
le laldo fumna ga nenkaa".
"cao" again seems excessive, but "ia" seems misused ("i certainly belive"?)
What would you suggest?
@PNotice that I use "nau" for scene changes, not for English "paragraphs"
(of which there are hundreds here).
".iza nau" seems quite adequate in Loglan, but the English translations
come out seeming brutal and jerky.
Is Loglan just more compact, or is the problem real?
In other places too the flow seems jerky sometimes.
Maybe I am inexpert in handling the Loglan.
Or maybe I am inexpert in writing short stories.
I felt in this one I had to cut and shorten to match the weight of

the material, and the patience of the audience, with the
words by which it was represented.
Maybe I cut too much.
@PI have great trouble translating "just" and "only".
For example, "...better than just adequate" = "gudbi le pu sira tsufi",
where "sira" = "sinera".
Any suggestions for improvement?
@PGiven sentences R .ucanoi S .i T .urau dui (or similar where
"dui" is replaced by some sentence including "tiu").
What is the referent of "dui"?
T: usually I don't want this one, as in this causal connected example.
S: usually this is what I want.
R .ucanoi S: this is the obvious choice from purely mechanistic
syntax rules, but it's hard to interpret with "dui" because it's connected.
@PIn a story like this, only the narrator can credibly use keks --
it's out of character for a teenage boy arguing with his mother
to use them even if he logli.
What afterthought forms would you use to get the effect of
kinumoi R ki kinurau S ki T?
R .umoi S .urau T means kinurau kinumoi R ki S ki T.
You may have noticed my "ci" forms, by analogy to metaphor:
R .umoi S .ucirau T, tightly binding S and T.
Somebody else already is using .icikou etc, but I am not sure
how they define "ci" forms.
>@ <EOF>

northwind.cantonese

Return-Path: LOJBAN%CUVMA.BITNET@mvs.oac.ucla.edu
Received: by luna.math.ucla.edu
    (Sendmail 5.57/1.07) id AA10297; Wed, 27 May 92 10:05:43 -0700
Received: from mvs.oac.ucla.edu by julia.math.ucla.edu via SMTP
    (Sendmail 5.61/1.07) id AA17925; Wed, 27 May 92 10:05:41 -0700
Message-Id: <9205271705.AA17925@julia.math.ucla.edu>
Received: from UCLAMVS.BITNET by MVS.OAC.UCLA.EDU (IBM MVS SMTP V2R1)
   with BSMTP id 6432; Wed, 27 May 92 10:05:06 PST
Received: (from cmsa.Berkeley.EDU for <LOJBAN@CUVMA.BITNET> via BSMTP)
Received: (from MAILER@UCBCMSA for MAILER@UCLAMVS via NJE)
 (UCLA/Mail V1.500 M-RSCS4554-4554-128); Wed, 27 May 92 10:04:44 PDT
Received: by UCBCMSA (Mailer R2.08 R208004) id 0300;
          Wed, 27 May 92 10:04:55 PDT
Date: Wed, 27 May 1992 12:46:23 EDT
Reply-To: John Cowan <cbmvax!snark.thyrsus.com!cowan@UUNET.UU.NET>
Sender: Lojban list <LOJBAN%CUVMA.BITNET@mvs.oac.ucla.edu>
From: John Cowan <cbmvax!snark.thyrsus.com!cowan@UUNET.UU.NET>
Subject: Re: taboo jadoo translation
To: Jim Carter <jimc>
In-Reply-To: <9205221150.AA06892@buphy.bu.edu>; from "John B Ross" at May 22,
              92 7:50 am

The "North Wind And Sun" story that has appeared in several versions is in

fact a Cantonese folk tale. In the interests of authenticity, I will
present it here in Cantonese with interlinear glosses in English and in
Mandarin (partial). Source: Yuan Jiahua et al, >Hanyu fangyan gaiyao<
[An Outline of the Chinese Dialects]. Peking: Wenzi gaige chuban-she, 1960.
I took this version from Ramsey, >The Languages Of China<.

Since Cantonese is scarcely a conlang, why present this text here? Primarily
because I distrust translations of translations. The text offered for
translation is obviously an English rendering of this very story, differing
only in minor details of wording from the colloquial translation printed by
Ramsey. I would suggest that all conlang translators give this text
(especially the English gloss :-)) the once-over to see if they missed any
salient points. I suspect the answer will be "yes", at least for Lojban.

ASCII transcription conventions:
    :   lengthens previous vowel
    '   aspiration
    "   umlaut in o", u"
    *   second changed tone (long high rising)
    CAPS   indicate Chinese categorizers

Disclaimer: I don't know any Chinese. I am responsible for any errors
in transcription.

        Pak7a fong1 t'ong2 yat8 t'au2
        bei3 feng1 tong2 ri4 tou2
        north wind together sun

Pak7a fong1 t'ong2 ma:i2 yat8 t'au2, yau4 yat7a ts'i5 hai2 su"5
bei3 feng2 tong2 ??? ri4 tou2 you3 yi1 ci4 ??? chu4
north wind together-with sun exist one TIME in process of

tsang1 lo"n6 k'o"i4 lo"ng4 ko5 tsi1 tsong1 pin1 yat7a ko5 pun3 si6
zheng1 lun4 ta1 liang3 ge zhi1 zhong1 ??? yi1 ge ben3 shi4
dispute he/they two ONES among which one ONE power

ta:i6. Sik7a tsik8 ko3 tsan6 si*, yau4 yat7a ko5 yan2, tso"k7b tsu"6
da4 shi4 zhi3 ??? zhen4 shi2 you3 yi1 ge ren2 zhao2 zhu4
great just then that PERIOD time exist one ONE person wear -ing

yat7a kin6 nu"n4 nu"n4 ke5 ts'o"ng2 p'ou* hai2 ko5 su"5 king1 kwo5.
yi1 jian4 nuan3 nuan3 de chang2 pao2 zai4 ??? chu4 jing1 guo4
one GARMENT warm warm 's long robe be at there pass by

K'o"i4 lo"ng4 ko5 tsau6 ts'ing2 yu"n3 lap8 yo"k7b, yu"2 kwo3
ta1 liang3 ge jiu4 qing2 yaun4 li4 yue1 ru2 guo3
he/they two ONES then be willing-to set-up agreement if

pin1 ko5 yau4 pun3 si6, nang2 sin1 sai3 tak7a ko3 ko5 yan2
??? ge you3 ben3 shi4 neng2 xian1 shi3 de2 ??? ge ren2
which ONE exist power can first cause get that ONE person

ts'o"i2 tso3 k'o"i4 ko3 kin6 p'ou*, tsau6 su"n5 pin1 ko5 ke5
chu2 ??? ta1 ??? jian4 pao2 jiu4 suan4 ??? ge ???

```
take off -ed his that GARMENT robe then be-regarded which ONE 's

pun3 si6 ta:i6.
ben3 shi4 da4
power great

Pak7a fong1 kam3 tsau6 yong6 tso"n6 k'o"i4 ke5
bei3 feng1 ??? jiu4 yong4 jin4 ta1 de
north wind thereupon then use exhaust he 's

lik8 lai2 ts'ui1 la5.  Sui2 tsi1 k'o"i4 yu"t8 fa:t7b ts'ui1 tak7a
li4 lai2 chui1 la shui2 zhi1 ta1 yue4 fa1 chui1 de2
strength come blow ! who know he the-more blow get

ka:u1 kwa:n1, ko3 ko5 yan2 yu"t8 fa:t7b la:m3 sat8 k'o"i4 kin6 p'ou*
jiao1 guan1 ??? ge ren2 yue4 fa1 lan3 shi2 ta1 jian4 pao2
intense that ONE person the-more pull tight his GARMENT robe

wai2 tsu"6 k'o"i4 ke5 san1 po5, tso"t7a tsi1 kin5 tak7a tou1 hai6
wei2 zhu4 ta1 de shen1 ??? zu2 zhi1 jian4 de2 dou1 shi
wrap -ing he 's body [pause] finally see get all is

m6 tsai5 tak7a kwo5 ke5 lo5, pak7a fong1 tsau6 m6 tsai5
bu2 zhi4 de2 guo4 ??? le bei3 feng1 jiu4 bu2 zhi4
not do -able ! north wind then not do

la5.  Ko3 tsan6 si* ko3 ko5 yat8 t'au2 tsau6 sai3 ts'o"t7a
la ??? zhen4 shi2 ??? ge ri4 tou2 jiu4 shi3 chu1
-ed! that PERIOD time that ONE sun then cause come-out

hou3 nu"n4 ke5 yit8 hei5.  Ko3 ko5 ha:ng2 lou6 ke5 yan2,
hao3 nuan4 de re4 qi4 ??? ge xing2 lu4 de ren2
really warm 's heat that ONE go road 's person

tsik7a hak7a tsau6 tso"ng1 k'o"i4 ko3 kin6 p'ou*
ji2 ke4 jiu4 jiang1 ta1 ??? jian4 pao2
immediately then taking he that GARMENT robe

t'u"t7b lak7a tso3.  Kam3 yo"ng3 ne1 ko3 ko5 pak7a fong1 tsau6
tuo1 ??? le ??? yang4 ne ??? ge bei3 feng1 jiu4
take off -ed this way [pause] that ONE north wind then

kam5 tsu"6 tsiu5 ying6, wa6 ko3 ko5 yat8 t'au2 hai6
??? zhu4 zhao4 ren4 hua4 ??? ge ri4 tou2 shi
force admit say that ONE sun is

lek7a kwo5 k'o"i4 tsi6 kei1 lo5.
??? guo4 ta1 zi4 ji3 le
capable [comparative] he himself!

--
cowan@snark.thyrsus.com     ...!uunet!cbmvax!snark!cowan
        e'osai ko sarji la lojban
```

northwind.engl

```
Return-Path: LOJBAN%CUVMA.BITNET@mvs.oac.ucla.edu
Received: by luna.math.ucla.edu
    (Sendmail 5.57/1.07) id AA04905; Tue, 26 May 92 18:45:02 -0700
Received: from mvs.oac.ucla.edu by julia.math.ucla.edu via SMTP
    (Sendmail 5.61/1.07) id AA11597; Tue, 26 May 92 18:45:02 -0700
Message-Id: <9205270145.AA11597@julia.math.ucla.edu>
Received: from UCLAMVS.BITNET by MVS.OAC.UCLA.EDU (IBM MVS SMTP V2R1)
   with BSMTP id 2763; Tue, 26 May 92 18:44:31 PST
Received: (from cmsa.Berkeley.EDU for <LOJBAN@CUVMA.BITNET> via BSMTP)
Received: (from MAILER@UCBCMSA for MAILER@UCLAMVS via NJE)
 (UCLA/Mail V1.500 M-RSCS8778-8778-67); Tue, 26 May 92 18:43:50 PDT
Received: by UCBCMSA (Mailer R2.08 R208004) id 6363;
          Tue, 26 May 92 18:44:06 PDT
Date:         Wed, 27 May 1992 11:43:17 +1000
Reply-To: nsn@MULLIAN.EE.MU.OZ.AU
Sender: Lojban list <LOJBAN%CUVMA.BITNET@mvs.oac.ucla.edu>
From: nsn@MULLIAN.EE.MU.OZ.AU
Subject:      North Wind tale
X-Cc:         nsn@ee.mu.oz.au
To: Jim Carter <jimc>



The following request appeared on conlang a while ago:

Translate the following paragraph into the constructed language of your
choice:

"The North Wind and the Sun were disputing(1) which was the stronger(2),
when a traveller came along wrapped in a warm cloak(3). They agreed that
the one who first succeeded in making(4) the traveller take his cloak(5)
off should(6) be considered stronger than the other(7). Then the North
Wind blew with all his might(8), but the more(9) he blew the more
closely did the traveller fold his cloak(10) around him; and at
last(11), the poor North Wind gave up the attempt(12). Then the Sun
shone out warmly(13), and immediately(14) the traveller took off his
cloak. And so the North Wind was obliged to confess(15) that the Sun was
the stronger of the two(16)."


This is my attempt:

la berbif. joi la sol. puki darlu lejei ri jikau ra vlimau le drata kei
co'i lenu lo litru vi klama gi'e tagji dasni lo kazgla kosta .i lego'i
cu tugni lenu le pamoi snada be lenu naldasri'a le litru le kosta du'o
ru'a vlimau le drata .ibazibo la berbif. cu rocrai brife .iku'i go ri
vlimau brife gi le litru cu tagmau vaungau le kosta ra .ibaze'e la
berbif. uu cu sisti lenu troci .ibabo la sol. cu glare dirce .ibazibo
le litru co'u dasni le kosta .iseki'ubo la berbif. cu bilga lenu tugni
ledu'u la sol. vlimau

and my word-for-word-er:
```

NorWind 'n' Sun argued [talking past now, son] on the truth of the sentence:
this one [connective disputed] that one is more powerful than the other,
at the instant when a traveller comes and tightly wears a warmth coat. They
agree that the first succeding in making-not-wear the traveller the coat
is, known to someone, [Postulate!] more powerful than the other. Then NorWind
was a most-effortful wind. But iff it was a more powerful wind, the traveller
tighter made-contain the coat him. Eventually NorWind [awwww] stopped trying.
Then Sun hot-radiated. Right after, the traveller stopped wearing the coat.
For that reason NorWind was obliged to agree to the sentence: Sun is more
powerful.

Comments solicited.

'Dera me xhama t"e larm"e,   T Nick Nicholas, EE & CS, Melbourne Uni
 Dera mbas blerimit | Mail: nsn@munagin.ee.mu.oz.au
 Me xhama t"e larm"e! | "Omiloume ellhnika/Esperanto parolata/
 Lumtunia nuk ka ngjyra tjera.' | {mika'e tavla baula lojban.je'uru'e}
 - Martin Camaj, _Nj"e Shp'i e Vetme_ | (Better .sig suggestions welcome)

reif.txt

A Sample of MEX in -gua!spi
Jim Carter <jimc@math.ucla.edu>, 5/18/90

In Loglan, and now in Lojban, a fairly elaborate syntax was proposed to
express mathematical expressions (MEX). I have prepared this sample of
MEX in -gua!spi to show one trivial solution, and I am posting it to
the list following <lojbab>'s encouragement to get discussion going.

On analysing the MEX problem I came to the conclusion that the basic
syntax of the language, or a substitute of substantially equal
function, was both necessary and sufficient to express MEX, and hence
when designing -gua!spi I omitted any special MEX syntax, rather than
inventing a separate but equivalent syntax for MEX.

However, I put some features into the core grammar specifically
anticipating the demands of MEX, such as default articles. Since the
tones mark where arguments start there is no syntactic need to have any
articles, and hence I caused every case of every primitive word to have
a default article, to be added to the occupying argument if it has none
of its own. Normally this is "xe" (pronounced "zhe"), corresponding to
Lojban "le", and the feature makes a -gua!spi sentence have noticeably
fewer words than the corresponding Lojban. Now for MEX, numbers are
defined, as many mathematicians do, as "X1 is a member of the
equivalence class of all sets with N members". For example, a biplane's
wings are a set, which are equivalent in count to the brothers Gemini,
etc. ad infinitum, and this set (class) of sets is "the number two".
Then the appropriate article for a case for a number, provided by
default on math functions, is "xu" ("all", "lea" in Old Loglan, changed
I think in Lojban), so the case occupant might mean "all pairs".

What follows is a translation of a short paragraph from a physics

textbook. It is hard to demonstrate how neat a certain feature is of
-gua!spi to listeners who do not know the language, and so I have
provided three translations: First, a (nearly) word-for-word
translation, then the original English, and finally a mechanical
translation with phrases marked. In the latter, [] encloses a sentence
(as in an abstraction), <> encloses a modal or subordinate clause, \\
encloses the anaphoric copy of the restricted phrase (internally
generated), and () encloses the antecedent of any other pronoun. The
main predicate of each argument has the case number appended ("vo" =
"to" is the predicate of an abstraction). The translator uses word
order and possessive apostrophes which are often correct in describing
life situations, but which get in the way of the mathematics.

The letterals and equations are written out as if spoken (analogous to
"forty two dollars and seventeen cents"); presumably in a real
mathematics text they would be written as letters.

One form lacking in the paragraph is a good, clear dimensioned
quantity, so I have added a difficult one at the end.

Perhaps the most useful thing you could do with this passage is to
translate it into Lojban using each of the competing MEX syntaxes, and
particularly, try using the core grammar with no MEX extension. You may
have to create ad-hoc case specifications, and I don't believe Lojban
has a word for "derivative", which you will have to jury-rig. There is
enough variety here to give you a broad, if brief, look at the
performance of each candidate.

```
\x{^:i !kun   !vn-qci|qky ^vu-zu-jio !se-xo-pse   |vu-gr-tfyn
   !fi-va-ga-xim !dvla-qfle ^fi-ve-fta !tl-co  /fi-plw !dvla-xble
   ^cmu !psla ^dvla-vzle  }
  {  Quantity energy hot absorbed-by some process anti-infinite
    (named           dQ)     by-rule    nbr.one    sum     dE
    product p      dV }
  { The heat absorbed in an infinitesimal process is given by the first
    law as dQ = dE + p dV.  (Actually d-bar Q; -gua!spi doesn't handle
    unusual letters yet.) }
{[argument energy's2 <hot \energy1\> <cargo/soak/ \energy2\ some
process4 <opposite infinity \process1\>> degree1 <performative name d Q2
\degree1\> plus all d E2 all all p's2 multiply3 all d V3 <standard
ordinal one1 \plus2\>]} %%


\x{^:i |vi-tl-co ^ja /zyn  !xo -kqer       ^dya     !kun   !qci   |qky
   ^kqa |gl-kri       ^fu-cmu    !jvyn   ^gzol /va-ga-xim !cfla-vzle }
  {     First    lets find some expression quotient amount energy hot
   big not increasing product    degrees moles named           cV }
  { Let us first obtain an expression for the molar specific heat cV
    at constant volume. }
{[imperative1 (all your2 (something's) set me3 (something)) discover
some mathematics2 to3 [(mathematics1) quotient2 all to's2 [the big1 <un-
property/increase/ \big1\> energy2 <hot \energy1\>] degree2 all all
kelvin's2 multiply3 all mole3 <performative name c V2 \quotient1\>]
```

```
<ordinal one \discover1\>]} %%


\x{^:o -sno              !cy !dvla-vzle }
  { Sufficient-condition zero    dV }
  { Then dV = 0 ...}
{[conjunction \discover1 11+22\ sufficient to2 [d V1 zero]]} %%


\x{^:o -sno              !can !so-vo-dem !jl ^vu-zu-dre-stl  !ci^cu^co
   !fu-psy |zu-plm !vo-dem !dvla-qfle ^dvla-xble }
  { Sufficient-condition change equation    (address    vector 5  2  1)
    simple example  equal      dQ         dE }
  { ...and (5.2.1) reduces simply to dQ = dE.  ("Example" is not very
    good; I need a word for "specifically".}
{[conjunction \sufficient1\ sufficient to2 [to [something1 same2
<object/address/ \same2\ to1 [five1 list all two2 all one3]>] change to2
[(to1) simple2 <set/example/ \simple2\ to1 [d Q1 same d E2]>]]]} %%


\x{^:o -sno              !dem !cfla-vzle |va-ga-zu-xim !zr-dya
   !vxln ^xbil    !qfle ^tfle ^vu-zo-kaw !gl-kri      -kqa
   !fi-zr-dya !vxln ^xbil !xble ^tfle ^vu-zo-kaw !gl-kri    -kqa }
  { Sufficient-condition equal    cV      name-of       quotient
    nu   derivative Q    T       status no-increase   size
    quotient   nu derivative E    T       status no-increase size }
  { Hence one obtains
             1 (dQ)    1 (dE)
       cV == - (--)  = - (--)
             v (dT)V   v (dT)V
    (v for nu, subscript V means constant volume.  See note below on -kri.}
{[conjunction \sufficient1 22+8\ sufficient to2 [c V1 <performative name
\c V2\ all nu's3 quotient1 all all Q's2 derivative2 T3 <object/state/
\derivative1\ to2 [to1 [(derivative1) big] un- increase]>> same all
nu's3 quotient2 all all E's2 derivative2 T3 <object/state/ \derivative1\
to2 [to1 [(derivative1) big] un- increase]>]]} %%


From Reif, Fred, "Fundamentals of Statistical and Thermal Physics",
McGraw-Hill, 1965, p. 156, "Specific Heats".


\x{^:i !vbna-xau       =spa     |vu    -xi     -tiw-fli /fi-kua |dya
   !xdem =co-cy-ku ^bzen }
  {     boat roundtrip to space when (typical) stop fly    fast quotient
   meter  1  0  7   second }
  { When landing the space shuttle's speed is 107 meters per second. }
{[space's2 boat1 and1 (which's1) roundtrip2 (space2) <typical actor/stop/
\and1\ to2 [(and1) fly]> fast <quotient all all one zero seven's2 meter2
\fast1\ all seconds3>]}


---- Appendix on -gua!spi ----
```

```
If you're trying to pronounce the examples, tones are:

  / rising      Starts or sometimes continues higher level phrase
  ! falling     Starts subphrase
  | down-up     Starts subordinate clause (so does vu, ve, vi, va)
  ^ up-down     Starts another subphrase at the same level
  = low-even    Compound word going in transitive case, usually X2
  - high-even   Compound word for abstraction case, or parallel cpd.

Phonemes differing from Lojban are:

  c ch      CHew
  j dj      John
  q sh      SHoe
  w ng      stroNG
  x zh      aZure, breZHnev
  y i (short)   thIck, Idiot
  : pause       hawai:i (this is a period in Lojban, but periods are
                 too hard to see in handwriting.)
  # schwa       Among

About kri-status in the sentence with the derivatives: the modal phrase
modifies xbil-derivative, and hence the status is that the derivative
doesn't change size.  In reality it should say that the volume doesn't
change.  But the volume of what?  It's the gas undergoing the
infinitesimal process assumed in the first sentence, but that gas never
appears as an argument, and since I'm trying to demonstrate a point,
I'm not going to "improve" the original English very much.  When you
have a parser-organizer breathing down your neck that will point out
every little inconsistency in your text, you begin to wonder if
ordinary people will ever accept a logical language.
```

short.msg

```
                  A Short Description of -gua!spi

I have never been satisfied with the mess which English makes of
grammar and semantics, and I was intrigued by Jim Brown's 1960 article
in Scientific American about Loglan.  However, Loglan doesn't go far
enough.  I implemented a number of suggestions from the Loglan
community to produce -gua!spi, which is based on Loglan but (from their
perspective) is radically different.

Morphology: C = (bcdfgjkpqstvxz:), V = (aeilmnoruwy), word = C^n V^n
with official assignments in CV, CCV, CVV, CCVV, CVVV.  : is a glottal
stop, used for transforming vowel-initial words into a CV^n form.  This
morphology is trivially resolvable into words.  Stolen foreign words
(like :au stralo pi te ku) tend to break up into syllables but the
compound word rules keep them together.  Each word has a Chinese-style
tone, represented by the symbols -/|!^=

Grammar: Words are divided into prefixes and predicates (plus sentence
start prefixes and two words to force phrase endings).  A phrase
consists of optional prefixes, a predicate (which may be several words
```

```
as a compound) and subphrases among the prefixes or after (not among)
the predicate.  The tones cue the start and end of (most) phrases.
Words can be grouped in phrases without any reference to their meaning.

Organization (again nearly meaning-independent):
   A.  A pronoun represents words, not the referent of words, and
is interpreted by copying the referent in place of the pronoun.  This
greatly simplifies semantic analysis.  By the time copying occurs,
though, showers of context have joined the original, so the copy will
be interpreted the same.  The effect is almost the same as if a pointer
to the original's referent had been copied.
   B.  One form of subphrase is a "modal phrase" for tenses, speaker
ID, listener, etc.  A stack is provided whereby these can be supplied
to phrases automatically, and can be changed and restored -- e.g. for
story dialog even with nested quotes, so you don't have to say "Joe
said" over and over.
   C.  Compound words are eventually split up so each predicate word has
its own subphrase; arguments are replicated as needed.

Vocabulary:  The Loglan - Lojban lexicon is the starting point for
-gua!spi, though I have added and removed a few words and have
extensively adjusted the word definitions to work well in compound
words.  There are about 1400 content words; any compound word can be
interpreted as a combination of these basis words.

Semantics: The predicates are interpreted by predicate calculus: each
predicate word represents a boolean (or fuzzy logic) function which is
true for thus-related arguments.  Argument sub-phrases fill cases in
order, up to five cases, except that certain optional prefixes act like
Latin case endings to override the natural order.  While the labels X1,
X2 etc. are used to designate the cases, X1 and X2 frequently act like
a traditional nominative and accusative case.  In arguments a (normally
unspoken, normally nominative) placeholder gives a "that which is..."
kind of interpretation; variants on this interpretation are available
through prefixes.  Related words have similar definitions, e.g. -ber =
X1 carries X2 to X3 from X4 via X5, and all transitive motion words
have a similar definition.  Any predicate can appear in any kind of
phrase (main, argument, or modal).

Further documentation available at this time:

guaspi.txt      A longer discussion of -gua!spi (65 Kbytes)
guaspi.sty      LaTeX style file to print it out
xankua.part1, part2, part3   Dictionary of -gua!spi, tab separated database
```

tonal.natlangs

```
Return-Path: cbmvax!snark!cowan@uunet.UU.NET
Received: by luna.math.ucla.edu
    (Sendmail 5.57/1.07) id AA05579; Mon, 27 Apr 92 09:39:27 -0700
Received: from relay1.UU.NET by julia.math.ucla.edu via SMTP
    (Sendmail 5.61/1.07) id AA22431; Mon, 27 Apr 92 09:39:25 -0700
Received: from uunet.uu.net (via LOCALHOST.UU.NET) by relay1.UU.NET with SMTP
    (5.61/UUNET-internet-primary) id AA25811; Mon, 27 Apr 92 12:39:26 -0400
```

```
Received: from cbmvax.UUCP by uunet.uu.net with UUCP/RMAIL
    (queueing-rmail) id 123803.16946; Mon, 27 Apr 1992 12:38:03 EDT
Received: by cbmvax.cbm.commodore.com (5.57/UUCP-Project/Commodore 2/8/91)
    id AA05518; Mon, 27 Apr 92 12:31:19 EDT
Received: by snark.thyrsus.com (/\==/\ Smail3.1.21.1 #21.19)
    id <m0lfXjx-0001vQC@snark.thyrsus.com>; Mon, 27 Apr 92 11:35 EDT
Message-Id: <m0lfXjx-0001vQC@snark.thyrsus.com>
From: cbmvax!snark.thyrsus.com!cowan@uunet.UU.NET (John Cowan)
Subject: Grammar by tones in natural languages
To: jimc (jim carter)
Date: Mon, 27 Apr 92 11:35:39 EDT
X-Mailer: ELM [version 2.3 PL11]



I posted the following message to linguist-list:

> Does anybody know of any languages in which tones are used not to discriminate
> lexical items but to specify syntactic relations?  In other words, are there
> any languages in which (to make up an example) >eat< means 'eat' and >dog<
> means 'dog' regardless of tone, but where:
>    eat1 dog1   means       'something eats and is a dog'
>    eat4 dog6   means       'an eater of dogs'
>    eat1 dog4   means       'the dog eats'
>    eat4 dog1   means       'the eater is a dog'
> and so on?

Note the choice of tones for the "made-up" examples.  :-)

The rest of this message is a summary of the replies I got.
It seems that -gua!spi style tones are unusual but not unknown in the world's
languages.  I would point out that the African-style tones are much simpler
than -gua!spi or Chinese, and are typically limited to high and low.

---------------

I received responses to my query about "grammar by tones" from the following:
    Tucker Childs <099CHILD@witsvma.wits.ac.za>
    Jeff Lansing <lansign@bend.ucsd.edu>
    Eric Schiller <schiller@sapir.uchicago.edu>
    Kathleen Hubbard <hubbard@garnet.berkeley.edu>
    Randy LaPolla <HSLAPOLLA@ccvax.as.edu.tw>
    John Goldsmith <gldsmth@bloomfield.uchicago.edu>
    John Kingston <KINGSTON@cs.umass.edu>
    Malcolm Ross <mdr412.coombs.anu.edu.au>
    Chet Creider <creider@cogsci.uwo.ca>
    <SABINO@ducvax.auburn.edu>
I wish to express my appreciation to all of these people for their kind
assistance.

Childs, Hubbard, Goldsmith, and Kingston cite examples from Niger-Congo,
mostly from Bantu languages.

Childs:
```

> In ... Sotho (Bantu), Kisi (Atlantic), it is usually verbs that lose
> lexical tone before nouns, and they are typically more heavily involved
> in grammatical tone (tense/aspect/mood distinctions).

Hubbard:

> In a lot of the Eastern Bantu languages, tone has an extremely
> low lexical load (i.e. there are only a few lexical minimal pairs
> distinguished solely by tone), and a high grammatical load
> (distinguishing one verb tense from another, etc.) I guess you'd
> say the morphological function is greater than the syntactic-
> relations function, but there are a few things: in Runyambo (NW
> Tanzania), there is a High Tone Deletion rule that applies in
> certain syntactic contexts, such as genitive phrases and verb-
> complement phrases. One thing the tonal pattern can tell you is
> whether the word following the verb is part of the same clause or
> rather a vocative or right-dislocated subject:
> /babona' kato'/   --> babona kato'   "they see Kato" (deletion)
> vs.
> /babona' kato'/   --> babona' kato'   "they see, Kato" (no deletion)
> /abona' kato'/ --> abona' kato'   "he sees, Kato does" (ditto)
> Another interesting deletion fact is that the rule normally does not apply
> between a noun and a following adjective, but it does when the sequence has
> become somehow "lexicalized". For instance, the phrase "omuka'zi muku'ru"
> means 'old woman', but if the H tone on the noun is deleted, "omukazi muku'ru"
> means 'eldest wife'. Sort of a compound. Likewise, "embu'zi m'bi" means
> 'bad goat', but when it appears in a well-known proverb it's "embuzi m'bi"
> -- with the H tone on the noun deleted -- as though in that context
> "bad-goat" is a lexical item.

Goldsmith:

> Igbo has quite fixed word order, so the subject will always precede
> the verb and the object will always follow it. But in a range of
> syntactic constructions, there is a rule of tonal mutation that will
> modify the tone of the object noun in much the way that you're asking
> for; and in a smaller number of constructions, there is a floating
> tone in between the subject and the verb which moves leftward,
> attaching to the noun and modifying its tonal pattern also.

Kingston:

> In Bakweri, a Bantu language of the Cameroon, the tone pattern on the verb
> in a relative clause differs between cases where the head noun is
> coreferent with the subject of the relative clause and that where it's
> coreferent with the object, so tone is, indirectly, related to the
> expression of grammatical relations.

Creider mentions something even closer to what I had in mind:

> All of the Southern Nilotic and most of the Eastern Nilotic languages
> use tone to signal syntactic case. E.g. (Kipsigis, Southern Nilotic)
> ke:re tE:tA (f-h l-h) (s/he is looking at the cow)

> see   cow
> ke:re tE:tA (f-h l-l) (the cow is looking at her/him)
> (Nandi, Southern Nilotic)
> ame se:se:nik (h-l h-h-f) (it eats dogs)
> eat dogs
> ame se:se:nik (h-h l-h-l) (dogs eat it)

(Interestingly, my only available information on Nandi classifies it as
Bantu/Niger-Congo, not Nilotic. Is this a slip, an error in my information,
a genuine dispute, or two languages with the same name?)

Schiller says that Hmong illustrates syntactic tone, but without details.
LaPolla refers to the use of tone to specify verb aspect in some Chinese
dialects. Ross refers to a case in Takia (Austronesian) from his own research,
but says it is still tentative. Sabino says that Virgin Islands Creole English
distinguishes future from negative by overall sentence contour, which doesn't
fit my paradigm (English distinguishes statement from ironic question by
contour, after all).

Finally, Lansing points to English:

> A _silver knife_ (low high) is a knife made out of silver, and a _silver
> knife_ (high low) is a knife for cutting silver.

Informal Bibliography (apologies for the lack of proper citations):

Martha Ratliff's article on Tonal doublets in LTBA (Linguistics of the
Tibeto-Burman area).

Kathleen Hubbard, an LSA paper in January on the syntax-phonology interactions
in Runyambo.

Green and Igwe's grammar of Igbo.

John Goldsmith's 1976 Ph.D. dissertation from MIT,
available from the IULC, entitled Autosegmental Phonology.

Orin Gensler at the linguistics department, University of California at
Berkeley; a master's thesis on Bakweri tone.

Sprauve, Gilbert. 1974. Towards a Reconstruction of Virgin Islands Creole
Phonology. Ph.D diss., Princeton University.

--
cowan@snark.thyrsus.com   ...!uunet!cbmvax!snark!cowan
        e'osai ko sarji la lojban