Notebook 3

THE PRESENT STATE OF THE LOGLAN™ LANGUAGE

by

James Cooke Brown

The Loglan™ Institute, Inc.
A Non-Profit Research Corporation
1701 Northeast 75th Street
Gainesville FL 32601
U. S. A.

CHAPTER 3

LEXICON (WORDS & SPEECH PARTS)

3.1. Definitions and Conventions: In this chapter the vocabulary of Loglan is partitioned into its 69 "parts of speech", or sets of grammatically interchangeable words. These sets are called **lexemes**, and we will consider them one at a time in the alphabetic order of their simplest or most representative members. These representative members, written in upper case letters, serve as **lexeme names.** We commence with Lexeme A, the Afterthought Connectives, and end with Lexeme ZO, the Quantity Abstractor.

If a lexeme has more than one member, these are called its **allolexes.** The allolexes of a lexeme are its equally permissible alternative expressions; they are the "interchangeable elements" of which that lexeme is composed. Lexemes which have only one member are called **monolexic.** In the section devoted to each lexeme, our object will be to show how its various allolexes are formed morphologically, how it is used by the grammar, and the range of meanings of its allolexes. Whenever possible, the list of allolexes is complete.

The allolexes of all the lexemes of a language are the **lexes** of that language. They are the words or phrases which, like English 'nevertheless', the grammar treats as a single word. In Loglan all lexes are words; that is, they are spoken pauselessly and written without internal spaces. The **lexer** is that part of the human central system, or of some computer program designed to process language, which identifies lexes as such, and which assigns them to their lexemes. In the system of computer programs that constitute the **machine grammar** of Loglan, the lexer is part of the **preparser** subsystem.

From the point of view of the grammar, each allolex of a non-monolexic lexeme is an equally legitimate occupant of whatever place any of them occupies in any utterance. Thus it makes no difference to the grammar which allolex of a lexeme a speaker chooses; for the grammatical structure of an utterance does not change when one allolex of a lexeme replaces another. So in a certain sense, if a learner has learned how to use one allolex of a lexeme, da has learned how to use them all. For they all have the same grammatical privileges. Thus, there are only semantical differences between allolexes of the same lexeme, not grammatical ones. It is in that sense that allolexes of the same lexeme are **grammatically interchangeable.**

The lexemes of Loglan are **exclusive.** That is, if a lex is a member of a certain lexeme, then it is a member of no other. This is not true of English, where the word 'bank', for example, belongs to at least three lexemes, each with distinct grammatical roles. The uniqueness of lexemic assignment in Loglan removes a major source of linguistic ambiguity from the language. This is the **lexemic ambiguity** ('They are flying planes') which seems to be both massive and universal in natural languages.

Lexemes whose names are 'M' followed by a numeral are called **M-lexemes** or **machine lexemes.** These and certain other machine-oriented lexemes do not appear in human Loglan and are visible and audible only to the machine. The human user of the grammar need not concern daself with them. The role of machine lexemes in the machine grammar is explained in the next chapter. The names of lexemes used only by the machine are *-ed in the section headings. A total of 17 lexemes are occasioned only by the machine's needs. So from the human user's point of view, there are only 52 lexemes in Loglan.

Nine lexemes, or about one out of six in the human lexicon, differentiate connective words from one another. This is an extraordinarily high proportion compared to any natural language. As a logical language, Loglan makes great use of the principle of **connectivity**, the principle by which the truth-values of several or many sentences may be related to one another in a single sentence.

The PREDA lexeme is the largest lexeme in the lexicon, containing about 80% of all dictionary entries. PREDA's are all the predicates of the language, that is, its noun-, verb- or adjective-like words as well as many of its adverbs and prepositions.

Many allolexes of the non-PREDA lexemes are compound structure words. The **compounding formulas** which show how these lexes are generated are composed of lexeme names, the sign of alternation [/], of concatenation [+], and of identity [=]. Thus 'NI/TAI + FI = UI' means that members of the NI (nee) or TAI (tigh) lexemes may be joined together with members of the FI (fee) lexeme to produce compound structure words which will be lexed as members of the UI (wee) lexeme.

The words 'operand', 'modificand' and 'connectand' are originally Latin words meaning 'that which is operated upon', 'that which is modified' and 'that which is connected', respectively; they are used freely throughout the text.

'R' followed by a numeral, e.g., 'R12', refers to a particular rule of grammar that will be found under that number in the next chapter.

The **parse** of an utterance is a specification of how the grammar generates or understands it. A parse may be partially shown by a full parenthesization of the utterance once it is stripped of any other punctuation. The nested parentheses show the order in which the lexes in the utterance are to be grouped. Thus the parse of **Da, a de, e di** = 'X or Y and Z' is partially given by **((da a de) e di).** This shows that the expression is to be understood by first grouping **da a de** together, and then grouping the result with **di.** Note that this structure is independent of the semantic values of **a** and **e**, which are in fact allolexes of the same lexeme (Lexeme A, the first lexeme in the lexicon). In this chapter, the production sign [=>] will sometimes mean 'is parsed as'.

Reference will be made throughout this chapter to various grammatical structures whose origins and functions will not be completely understood until they are studied in the grammar. In general, the lexicon acquires its structure from the grammar while the grammar acquires its semantic variety from the lexicon. In truth they are inseparable; neither can be deeply studied without reference to the other.

Lexeme A: Afterthought Connectives (Eks)

These are the afterthought connectives used between predicates, arguments, linked arguments, argument modifiers, and sentence modifiers. They are said to be "afterthought" because they are left-grouping, e.g., **da, a de, e di => ((da a de) e di).** So connectands may always be added to an ekked string without disturbing previously understood meanings.

When used between predicates, eks are prefixed with M11 by the preparser; when connecting linked arguments, the preparser inserts M1; when connecting argument modifiers, M6; and when connecting sentence modifiers--called simply "modifiers" in the grammar--M5. Only when used to connect arguments are eks unmarked.

The complete list of simple eks is:

a	...and/or...
e	...and...
o	...if and only if...
u	...whether...
nuu	whether..., ...
noa	...only if...
noe	not...but...
nou	not...whether...
anoi	...if...
enoi	...but not...
onoi	...or...but not both
nuunoi	whether..., not...
noanoi	not both...and...
noenoi	neither...nor...

In addition, any member of the PA-Lexeme may be appended to an ek, producing, for example, tensed, located or even "motivated" connectives: **efa** = 'and then'; **evi** = 'and here (at this place)'; **emoi** = 'and with this intention'; and so on. All such compounds are treated grammatically as members of A. Not many of these A + PA compounds have been explored, however; their semantic domain seems to transcend any found in the natural languages. The loglanist is invited to explore this new country. See Lexeme PA for a list of the PA components of A + PA compounds.

Lexeme ACI: Hyphenating Eks

These are any member of the A-Lexeme suffixed by **-ci**: thus **aci, eci, apaci, noanoici**, and so on, are all hyphenating eks. ACI connectives have an effect analogous to that of hyphen **ci** in a predicate string (see Lexeme CI); that is, they consolidate the two adjacent elements into a single connectand. At the moment, ACI is confined grammatically to ekking arguments and predicates; among the latter, the preparser will have prefixed M9 to the ACI word. The use of hyphenating eks could, of course, be extended to other ekked structures in the grammar, for example, to ekked modifiers. It would be grammatically costly to do so, however, since introducing ACI to an ekking structure usually involves the addition of 2 to 4 rules, and a new M-lexeme will always be required for every structure so accommodated. So for the present, and until a use for them in other contexts can be demonstrated, hyphenating eks are confined to connecting the two main ingredients of a sentence, arguments and predicates.

Lexeme AGE: Right-Grouping Eks

These are any member of the A-Lexeme suffixed by **-ge**: thus **age, ege, apage, noanoige**, and so on, are all right-grouping eks. AGE connectives have an effect analogous to the effect of the grouping operator **ge** in a predicate string; that is, they consolidate the entire right portion of a string of ekked elements into a single connectand. At the moment AGE words are confined grammatically to ekking arguments and predicates; among the latter, the preparser will have prefixed M11 to the AGE word. The use of right-grouping eks could, of course, be extended to other ekked structures in the grammar, for example, to ekked modifiers. It would be grammatically costly to do so, however, since introducing AGE to a new ekking structure usually involves the addition of 2 to 4 rules, and always involves the addition of another M-lexeme. So for the present, and until a use for them in other contexts can be demonstrated, right-grouping eks are confined to arguments and predicates.

Lexeme BI: Identity Operators

These are the "little word predicates" which must be kept grammatically separate from the PREDA-Lexeme because their compounds are recognized by the preparser. If it weren't for this morphological function, BI and kin would be members of PREDA. The current list of BI words is

bi	is equal/identical to
bie	is a member of
biu	is part of
cie	is less than
cio	is greater than

and the compounds

ciebi	is less than or equal to
ciobi	is greater than or equal to

Other compounds may be made by prefixing **no-** or **nu-** to any of these with obvious meanings. Thus **nocio** makes the same claim as **ciebi.** No doubt mathematicians will have other allolexes to add to BI,

BI is an open lexeme; new members may added at any time.

*Lexeme BAD

This is special lexeme used by the lexer as a bin for words it doesn't recognize. Human users may also have such a bin for unlexed words; but unlike the machine, humans usually guess what lexeme unknown words belong to.

Lexeme CA: Predicate Word Connectives (Sheks)

These are the connectives used to join individual predicate words in an afterthought, i.e., left-grouped, mode. Morphologically, each shek is the result of inserting **c** before the characteristic vowel of an ek (see Lexeme A). The current list of sheks is

ca	...and/or...
ce	...and...
co	...if and only if...
cu	...whether...
nucu	whether..., ...
noca	...only if...
noce	not...but...
nocu	not...whether...
canoi	...if...
cenoi	...but not...
conoi	...or...but not both
nucunoi	whether..., not...
nocanoi	not both...and...
nocenoi	neither...nor...

Lexeme CI: The Interverbal Hyphen

CI is monolexic; its one allolex, **ci**, therefore has a unique role in the language. **Ci** is used between pairs of predicate units--usually, individual predicate words--to make larger predicate units inside predicate strings. Such "hyphenated" units then function as single modifiers or modificands in the predicate string.

Lexeme CUI: The Shek Left-Parenthesis

CUI is also monolexic; its one member, **cui**, is the left-parenthesis inside predicate strings. It is used with sheks (see Lexeme CA) to increase their leftward scope beyond the single predicate word which is the normal left-connectand of a shek. No right-parenthesis is necessary in predicate strings.

Lexeme DA: Variables

The allolexes of DA are found in six phonemically distinct series: (1) the five replacing, or 3rd person, variables **da de di do du** ('X' 'Y' 'H' 'W' 'Q'; linguists call the use of such short expressions to replace longer ones "anaphora", hence these are the anaphoric variables); (2) the four non-designating variables **ba be bo bu** ('x' 'y' 'w' 'q'), the "bound variables" of the logician); (3) the three 1st and 2nd person variables **mi tu mu** ('I/me', 'you', 'we/us', the latter in the sense of 'you and I/me jointly'); (4) the mixed 1st and 3rd person variables **mua mue mui muo muu** (the other sense of 'we/us': 'I/me and X/Y/H/W/Q jointly'); (5) the mixed 2nd and 3rd person variables **tua tue tui tuo tuu** ('you and X/Y/H/W/Q jointly'); and (6) the three pairs of demonstrative variables, namely **ti ta** ('this' vs. 'that' as applied to non-linguistic objects), **toi toa** ('this (most recent) utterance' vs. 'that (earlier) utterance'), and **tiu tau** ('the situation referred to by this (most recent) utterance' vs. 'the situation referred to by that (earlier) utterance'). The last two pairs of demonstratives do not, so far as we know, have parallels in natural language. But the distinctions they make will be found to be indispensable to the user of a logical language.

Lexeme DIO: Argument Tags

Argument tags are optional prepositions that can be selectively used before any argument, but are especially useful in three situations: when (1) the argument so tagged is out of its usual order, as may be required during translation, for example, to match some natural word-order; (2) the tagged argument is a sufori argument of its predicate (experience has shown that the meanings of the first 3 places of a predicate are easy to remember, but that if a predicate has sufo (four or more) places, the meanings of the sufori places are difficult to remember); or (3) the auditor is suspected of not knowing the place structure of that predicate, as is often the case, for example, when the speaker is a teacher and the auditor is a learner.

Loglan argument tags are of two types: (a) the 11 optional case tags, and (b) the five ordinal tags once called "HB-tags" [for "Hixson-Bonewits", the nom-de-plume of the loglanist who first proposed them]. The two systems are discrete; they will seldom if ever be used together in the same utterance.

The 11 optional case tags, together with their source words and meanings, are as follows:

dio	(dirco)	'to/for'	Destinations or Beneficiaries
duo	(durzo)	'by'	Actors, Agents, Doers
foa	(folma)	'in/of'	Wholes, Sets, Collectivities
fua	(fruta)	'-'	Outputs, Purposes, Products
koe	(konte)	'by/for'	Quantities, Amounts, Values
mau	(cmalo)	'than'	Lessers in greater/lesser than rel'ns
neu	(nerbi)	'under'	Conditions, Fields, Circumstances
piu	(pisku)	'in/on'	Parts, Passives, Properties
roa	(groda)	'than'	Greaters in greater/lesser than rel'ns
sau	(satci)	'from'	Sources, Origins, Reasons, Causes
veu	(vetci)	'by/via'	Deeds, Means, Routes, Events, States

Studies have shown that these eleven cases accommodate all the places of the 800-odd primitive predicates; see List 4, Cases.

Morphologically, each case tag is derived from a Loglan primitive predicate meant to be of mnemonic value in learning its meaning, e.g., **groda** for "greaters". Tags are derived by a procedure similar to that used to derive CVV-form affixes from their primitives, but with this difference: if the result of applying the affix-deriving procedure produces a letter word (all forms ending in **-ei -ai -oi** or **-ao** are letter words; see Lexeme TAI), the letter **-u** must be substituted for the final **-i** or **-o** of that letter word. Four tags on the current list show the **u**-substitution: **mau, neu, sau** and **veu.** All case tag assignments are still provisional. The Institute would welcome suggestions before GPA (The Institute's acronym for 'Going Public Again') for alternative tags that have plainer derivations, or which have been drawn from primitives with more useful mnemonics.

It is believed that the eleven cases that these tags represent constitute a (nearly) complete set in the sense that the places of (nearly) any future Loglan predicate will be capable of being assigned some ordered subset of these 11 cases in such a way that no two places of that predicate will be assigned the same case unless the occupants of those places can be exchanged without altering the truth-value of any claim made with that predicate. This criterion is more complex to read about than it is to use. For example, the first two places of **ciktu** = '...is equal to...in dimension/feature...' are currently assigned the **Foa** Case, the entire string of case assignments being **foa foa piu** (Whole Whole Part). The dual assignment of the **Foa** Case to two places of this predicate reflects the fact that any two occupants of these places in a true sentence may be exchanged without rendering the resulting sentence false. Thus if X is equal to Y on dimension W, then Y is equal to X on that dimension.

The English prepositions in the third column above are meant only to suggest how these tags may occasionally be translated into English. They are not definitions of the case tags.

The second system of argument-tagging--an historically earlier one, in fact--uses five ordinal tags to identify nothing more than the number of the tagged argument's normal position in the place structure of its predicate; thus **zua zue zui zuo zuu** mean 'the normally 1st', 'the normally 2nd', '3rd', '4th', or '5th argument of its predicate', respectively. [Since they were proposed and adopted in 1977, the ordinal tags had been assigned the morphemes **pua pui pue puo puu**; they were shifted to their present /zuV/ values in 1987 to accommodate a useful development of the tense system; see PA.]

It is very likely that we will not need both tagging systems. One will almost certainly be found to be enough. We have installed them both in this still-experimental version of the language in order to discover which tagging system will be used preferentially. As usual, we will let usage decide.

Lexeme DJAN: Name Words

These are all the C-final words found by the lexer. Names are used in four places in the grammar: they are formed into strings by R68-9 (**Djan Pol Djonz**); they are made into designations by R80, which precedes them with **la** (**La Djan Pol Djonz, pa pasko merki mursi ci kapta** = 'John Paul Jones was an early American sea-captain'); they are made into vocatives by R177, which puts **Hoi** in front of them (**Hoi Djan Pol** = 'O John Paul!'); and they are used as unmarked vocatives by R179 (**Pol, gotsi** = 'Paul, go!').

*Lexeme END

This machine-oriented lexeme has exactly one allolex, the period or full stop [.]. This is a special, and probably temporary, concatenation sign used by the machine to prepare a specimen composed of 2 or more utterances for sequential parsing.

*Lexeme FI: The Utterance Ordinal Suffix

This monolexic lexeme is used only by the preparser to recognize the NI/TAI + FI compounds. These are the "utterance ordinals" which are members of Lexeme UI, for example, **Nefi** and **Rafi** ('Firstly' and 'Finally'). The suffix **-fi** is incidentally also used to form the lower-case Greek vowel letter-words **afi efi ifi**, etc.; but the recognition of letter-words is handled by the lexer directly and does not require **-fi** as a separate lexeme; see TAI for other letter-words.

Lexeme GE: The Grouping Operator

Ge is the only member of its lexeme; it serves to make right-extending groups within predicate strings. Like all strings of similar elements in Loglan, predicate strings group left whenever they are unmarked: (i) **(((preda preda) preda) preda).** **Ge** functions as a left-parenthesis, sometimes matched with an optional right-parenthesis **gue**, and interrupts that normal pattern by turning all that follows **ge**, or all that lies between **ge** and **gue** in case there is a **gue**, into a single modificand; see Lexeme GUE. Thus (ii) **(preda ge ((preda preda) preda))** is a **ge**-marked string. Notice that the normal right-grouping pattern resumes in the portion of the string that follows **ge**. **Ge** is often well-translated by the English phrase 'for a'; for example, 'That's big for a little woman's suitcase' = **Ta groda ge cmalo fumna racbao.** The predicate in this utterance has the same structure as (ii).

Lexeme GI: The Fronting Operator

GI has two allolexes, **gi** and **goi**. **Gi** allows the "fronting" of strings of possibly arguments and/or modifiers, that is to say, it permits the speaker to move all or some of these items to the front of the sentence from their Loglan-normal position at the rear. **Gi** thus permits Loglan to be spoken in the O-S-V (Object-Subject-Verb) word-order by appearing between Object and Subject (or between Object and Verb, in the case of imperatives). There is usually no English word or phrase corresponding to **gi** in the O-S-V sentences which it helps in this way to translate. For example, in the sentence 'In May to Europe, John travels' = **Na la Femen, dio la Europas, gi la Djan, traci** Loglan **gi** corresponds to nothing in the English sentence except the comma. **Gi** is in this sense one of Loglan's "spoken punctuation marks". This same sentence in Loglan-normal order is **La Djan, traci la Europas, na la Femen.** Note that the case tag (preposition) **dio** (see Lexeme DIO) is left out in this minimally marked normal word order.

It is not necessary to front the entire string of normally trailing arguments and/or modifiers when using **gi**; only some of them need be shifted. But if only one is shifted, it must be marked by **gi**. Thus **Na la Femen, gi la Djan, traci la Europas** = 'In May, John travels to Europe' also requires **gi**, and corresponds to a much more common English word order.

GI's second allolex, **goi**, permits the objects called "prenex quantifiers" in the argot of logicians to be identified as such. Prenex, or sentence, quantifiers are the strings of quantified variables that precede a sentence in which these same variables appear in unquantified form; e.g., 'For every x there is a y such that y is faster than x' = **Raba be goi be kukra ba.** Thus **goi** may often be translated by English 'such that'.

Lexeme GO: The Inversion Operator

GO is monolexic; it is the predicate inversion operator. **Go** is used in a predicate string to exchange the positions of the right and left segments formed by some partition of that string. The segmentation of the original string must coincide with some modifier-modificand joint in it. Thus (i) **Da mutce groda mrenu** may be segmented between **groda** and **mrenu**, and inverted as (ii) **Da mrenu go mutce groda** = 'X is a man who is very big'; but (i) may not be segmented between **mutce** and **groda** and inverted as (iii) **Da groda mrenu go mutce** = *'X is big man who is very'. (iii) is a legitimate inversion, but of another Loglan sentence: (iv) **Da mutce ge groda mrenu** = 'X is extreme for a big man'. Thus inversion with **go** sometimes renders some of the punctuation of the original sentence redundant, in this case **ge**, and so omissible. Thus (v) ***Da ge groda mrenu go mutce** means the same thing as (iii)--and as (iv), for that matter--and parses as essentially the same structure; but the redundant **ge** is not considered good usage, and so I have *-ed the utterance in which it appears.

Lexeme GU: The Optional Right Boundary Marker ("Comma")

GU is also monolexic and is the optional comma-word. It is used to mark the right boundaries of structures whose boundaries would not be clear without it. **Gu** is never used redundantly, however, i.e., when the boundary with which it coincides is already clear. **Gu** is sometimes, but not always, replaceable by a literal comma [,] in text or by a pause in speech. One must know the particular grammar rules involved in this and other uses of **gu** to know whether such substitutions are admissible. In general, the cases in which **gu** may be replaced by pause/comma far outnumber the few cases in which it may not.

Lexeme GUE: The GE-Group Optional Terminator

GUE is monolexic and, like GU, optional: that is, **gue** may always be omitted when it isn't needed. **Gue** has only one function: to terminate a **ge**-group when that group does not run to the end of the predicate string. Thus, **gue** is always matchable with a **ge** but not all **ge**'s have matching **gue**'s; see Lexeme GE.

Lexeme HOI: The Vocative Marker

Also monolexic, HOI is the left or leading mark of some vocative expressions. If the vocative is a name word (Lexeme DJAN) or a string of name words (**Djan Pol Djonz**), leading **Hoi** is optional unless it is needed to separate a vocatively used name from a preceding name...perhaps one used designatively: **Donsu ti la Djan Pol Djonz, Hoi Pit** = 'Give this to John Paul Jones, O Pete!' Here **Hoi**--in which the H is always capitalized in text--serves, like English 'O', to separate the vocative name from the designative one. Without it, **Donsu ti la Djan Pol Djonz Pit** says 'Give this to John Paul Jones Pete', which designates a person with a tetradic name. So **Hoi Pit** is obligatory if Pete is to be successfully instructed to be the donor. **Hoi** is optional in front of names which need no separation from the prequel: **Donsu ti le ditca, Pit** = 'Give this to the teacher, Pete'; for in such contexts it is clear where the vocative name begins.

A second use of **Hoi** is to mark descriptions used vocatively, such as **Nenkaa Hoi Ganbra** = 'Come in, O Noble One!' Here the **Hoi** is obligatory. Without it the invitation becomes **Nenkaa ganbra** = 'Be in-comingly noble!' In text, the initial letters of all the predicate words in vocative descriptions are capitalized.

Lexeme HU: The Interrogative Argument

HU is monolexic; its one member, **hu,** is the interrogative argument. That is to say, **hu** makes possible the kinds of questions to which arguments are answers. **Hu** may be spoken alone or take the position of some argument in an utterance. For example, **Hu hijra** asks 'Who is present?' Just **Mi,** or **La Djan**, or **Mi hijra** are all possible answers...some, admittedly, more useful than others.

HU is also used by the compound-lexer to identify **Nahu**-type compounds. The most common of these are **Nahu** = 'When?', **Vihu** = 'Where?' and the four kinds of Loglan 'Why?': **Kouhu** = 'Because of what cause?', **Moihu** = 'Because of what motive, or with what intent?', **Rauhu** = 'Because of what reason, or with what justification?' and **Soahu** = 'From what premise(s)?' or 'With what logical justification?'. All these PA + HU compounds are members of the Lexeme UI; see UI for the complete list. If it weren't for this use of HU by the preparser, **hu** would be just another allolex of Lexeme DA; q.v.

Lexeme I: Sentence Connectives (Eesheks)

I is the principal sentence connective, but there are many others. Almost any compound which has I- as its leading element (except ICI and IGE words; see below) is an eeshek. Thus, there are I itself and the 14 eesheks formed by preceding a shek with I-:

I	... .
Ica	...; or..., and possibly both
Ice	...; and...
Ico	...if and only if...
Icu	...; whether or not...
Inucu	whether...; ...
Inoca	...only if...
Inoce	not...but...
Inocu	not...whether...
Icanoi	...if...
Icenoi	...; but not...
Iconoi	...; or..., but not both
Inucunoi	whether..., not...
Inocanoi	not both...; and...
Inocenoi	neither...nor...

In addition, any of these 15 eesheks may be compounded with a following PA word, just like any ek may be; and these, too, are members of the I-Lexeme. But the eeshek that is usually used in such compounds is I itself. If the PA word is a temporal, the result is a tensed connective (e.g., **Ifa** = 'And later...'); if the PA is a spatial, the result is a located connective (**Ivu** = 'And far away...'); and if the PA word is a causal operator--these are a well-defined subset of the PA-Lexeme--the result is one of the afterthought causal connectives. These are important for translating from, or to minds trained within, the Indo-European languages, so I give the complete set of 16 afterthought causal connectives here. Grammatically, the important thing to remember is that each of these I + PA words is a member of the I-Lexeme and treated just like any other eeshek:

Ikou	Effect	E because of cause C.
Inukou		C; therefore E.
Inokou		E although C.
Inunokou		C; nevertheless E.
Imoi	Act	A because of motive M.
Inumoi		M; therefore A.
Inomoi		A although M.
Inunomoi		M; nevertheless A.
Irau	Decision	D because of reason R.
Inurau		R; therefore D.
Inorau		D although R.
Inunorau		R; nevertheless D.
Isoa	Conclusion	C because of premise(s) P.
Inusoa		P; therefore C.
Inosoa		C although P.
Inunosoa		P; nevertheless C.

Thus, the category of sentence connectives in Loglan is extremely varied; for example, **Inocenoipacenoina** (Inocenoi + pacenoina) is one of its potential members...although it is mind-boggling to even think of what it might mean. Clearly this is a domain of logical possibilities that far exceeds the naturally sayable. Loglanists are invited to explore this vast semantic domain and make what use of it they can or will.

Lexeme ICI: Hyphenating Eesheks

These **Ici** and **Icaci**-type words are recognized by the compound-lexer. They are eesheks in all their wonderful variety but adorned with a trailing **-ci** that makes them hyphenating. The hyphenating eesheks are a special variety of sentence connectives which have the same effect on a string of connected sentences as ACI-words have on a string of ekked arguments or predicates, namely, they "hyphenate" or close-bind a pair utterances in a string of connected utterances, making a single connectand out of them.

Lexeme IE: The Identity Interrogative

Ie, the single allolex of this lexeme, is the interrogative particle by which identity questions are raised. Given any sort of designation--a name, for example, or a description, or a variable--**ie** may be prefixed to it; and this converts the argument, or actually the sentence in which the argument is embedded, into a question which asks about the identity of the designatum of that argument. Thus, **Ie la Djan** means 'Which John?', **Ie le mrenu jia pa kamla lepo foatci** = 'Which man that came to dinner (i.e., to the "formal eating")?', or **Ie tu pa godzi go trena** = 'Which (of) you went by train (took the train)?' Thus **ie** has roughly the sense of English 'which...' or 'which of...' as applied to some set of possible identities.

Lexeme IGE: Right-Grouping Eesheks

This variety of sentence connectives is formed by attaching to any eeshek a trailing **-ge**; thus **Ige** and **Icage** are right-grouping eesheks, and are recognized as such by the compound-lexer. IGE is a special variety of I-connective which has the same effect as AGE has among connected predicates or arguments. In a string of connected utterances, it groups all that follow it into a single connectand.

Lexeme JE: The First Linking Operator

Je, which is grammatically unique, is the first linking operator. It is used to attach 2nd arguments to predicate words: thus, **Le farfu je le botci** = 'The father of the boy' and **Ta kukra je lo litla, grobou** = 'That's a faster-than-light ship'. See JUE for links to suteri arguments.

Lexeme JI: Argument Modification Links

At present there are five of these operators: **ji ja jii jie** and **pe.** All of them attach either modifiers or other arguments to arguments. In this way they accomplish "local modification" as opposed to the sentence-wide or "adverbial" role that unattached modifiers perform.

Ji is the identifying link (**Le mrenu ji vi le hasfa** = 'The man in the house'); **ja** is the predicating link, i.e., it gives incidental information about the designatum (**Le mrenu ja le ditca** = 'The man, who is incidentally the teacher'); **jii** is the identifying membership link (**Le mrenu jii le merka** = 'The man who is one of the Americans'); **jie** is the predicating membership link (**Le mrenu jie le brudi** = 'The man, who is incidentally one of the brothers'); and finally **pe** which is the postfixing genitive operator (**Le bukcu pe le ditca** = 'The book of the teacher'). **Pe** provides an alternative to the prefixed genitive (**Le le ditca, bukcu** = 'The teacher's book') just as in English, which is also a language in which both forms of the possessive exist.

JI is an open lexeme; other allolexes may be added from time to time.

Lexeme JIO: Subordinate Clause Links

There are currently two of these words, **jia** and **jio**; both are used to attach subordinate clauses to arguments. The two JIO words differ grammatically from JI words in that the operands of JIO words are sentences-- although such "sentences" may be single predicate words, i.e., "imperatives"-- while the operands of JI are either arguments or modifiers. Thus, **Da jio prano** means 'The X who runs'. As in the case of JI words, the distinction between the identifying and predicating senses of subordinate clauses is scrupulously maintained. Thus **Jio**-clauses identify; they correspond to the restrictive clauses which are usually not set off by commas in English text: **La Djan, jio prano ga blonda** = 'The John who runs is blond'. In contrast, **jia**-clauses predicate; they correspond to the non-restrictive clauses which are usually set off by commas in thoughtfully composed English text: **La Djan, jia prano ga blonda** = 'John, who incidentally runs, is blonde.' Notice how, with restrictive clauses, the translation into English actually assigns a word to Loglan **la**: 'The John who runs is blonde'. With non-restrictive clauses made with **jia**, Loglan **la** has, as usual, no translation: 'John, who incidentally runs, is blonde.' The English word 'incidentally' is one of the few unequivocal ways of showing that an English clause is non-restrictive. In Loglan the linking word itself carries that meaning.

Lexeme JO: Metaphorizers

Jo and kin are the metalinguistical operators by which a loglanist, if speaking or writing carefully, can call attention to whatever portion of da's speech is non-literal. **Jo** itself signifies that the preceding word was used metaphorically. **Tojo** means that the two preceding words were used metaphorically; **tejo**, that the three preceding words were so used, and so on. **Rajo** anywhere in an utterance means that the utterance as a whole is not to be taken literally.

Lexeme JUE: The Second Linking Operator

Jue, the sole member of its lexeme, is used in conjunction with **je** to link 3rd and subsequent arguments to predicate words; see JE. For example, if one wanted to designate a ship by alluding to the fact that it was not only faster than light (**kukra je lo litla**) but faster than light by 20 kilometers a second, one could say **Le kukra je lo litla jue lio tonikeimeikuasei, grobou** ('The faster-than-light by twenty-kilometers-per-second ship'). Somewhat less fancifully, **Le farfu je to botci jue la Meris** designates the father of two boys by (or through) Mary.

It is grammatical but bad usage to link sutori arguments to the predicate of a sentence with **je** and **jue**: ***Da farfu je to botci jue la Meris.** Equally understandable and better usage is the simpler preposition-free form **Da pa farfu to botci la Meris** = 'X fathered two boys through Mary'.

Lexeme KA: Prefix Members of Forethought Connectives (Keks)

Keks are separated pairs of connective words, like English 'Either...or...', which are used to make forethought connections between a wide variety of elements. Connections of this kind are said to be "forethought" in that the speaker must decide what kind of connection da is going to use before mentioning the elements which are to be connected.

KA and the next lexeme, KI, are two of the most widely distributed lexemes in Loglan grammar. They are used with M2 in R13 and R19 to kek links and linkargs; with M3 in R37 and R150 to kek predicates; and without any advance marking at all to kek modifiers in R63, arguments in R106, termsets in R125 and sentences in R160.

Each kek is composed of a prefix member chosen from the KA Lexeme and an infix member chosen from the KI lexeme, q.v. Thus **Ka da ki de farfu** is a sentence with a kekked 1st argument and means 'Either X or Y, and possibly both, are fathers'. Like the English expressions 'Either...or...', 'If...then...', 'Both...and...' and 'Neither...nor...', the earliest or "prefix" member of each pair--in this case **Ka**--announces that a connection is about to be made. **Ka** also specifies the kind of connection it is going to be, namely a logical alternation. The prefix element is then followed by the left-connectand, which is in turn followed by the infix member of the connective pair--in this case **ki**--and the connection is then completed by the right connectand. Thus it takes a minimum of 4 elements to make a kekked connection...5 if the KA element must be marked by either M2 or M3 for the machine.

There are two main series of keks. The first are the forethought logical connections like English 'Either...or...'; the second are the forethought causal connectives like English 'Because..., (therefore)...'. The logical series is semantically more fundamental and will be described first.

The Logical Keks: Morphologically, the prefix portion of a logical kek is formed by inserting a /k/ before the characteristic vowel of an ek; see the A Lexeme. The infix portion is then either **ki** or **kinoi** depending on whether the corresponding ek does not or does end with **-noi.** To show this relationship between eks and logical keks, here is the complete list of logical keks shown alongside eks of the same meaning:

a	ka...ki...	either...or...and possibly both
e	ke...ki...	both...and...
o	ko...ki...	if and only if...then...
u	nuku...ki...	#...whether...
nuu	ku...ki...	whether..., ...
noa	kanoi...ki...	if...then...
noe	kenoi...ki...	both not...and...)
nou	nukunoi...ki...	#not...whether...
anoi	ka...kinoi...	#...if...
enoi	ke...kinoi...	both...and not...
onoi	ko...kinoi...	either...or...but not both
nuunoi	ku...kinoi...	whether..., not...
noanoi	kanoi...kinoi...	either not...or not...and possibly neither
noenoi	kenoi...kinoi...	neither...nor...

Notice that the transformation of ek into kek is irregular in the case of the four independence connections: **u nuu nou nuunoi.** Also, I have marked with '#' as possibly misleading the English translations of a different four Loglan connections which have no forethought renderings in English.

The Causal Keks: Morphologically, the prefix element of a causal kek may be formed by removing the leading /i/ from a causal eeshek (see Lexeme I) and adding a final /ki/: thus **kouki** is obtained from **Ikou,** an I-word. A more direct way of describing these compound members of KA is to say that they are PA + KI when the PA word is a causal. (Temporal and spatial keks have not yet been defined; but they are certainly possible.) Here is the list of the causal keks paired with eesheks of the same meaning. Notice, however, that the eeshek used to derive the kek is the one that precedes that kind of causal element (i.e., cause or effect) in an afterthought construction. In the following lists, E = Effect where C = its Cause; A = Action where M = its Motive; D = Decision where R = the justifying Reason; and C = Conclusion where P = the supporting Premise.

E Ikou C	Nukouki E ki C	*Therefore E because of C.
C Inukou E	Kouki C ki E	Because of C, E.
E Inokou C	Nunokouki E ki C	*Nevertheless E although C.
C Inunokou E	Nokouki C ki E	Although C, E.
A Imoi M	Numoiki A ki M	*Therefore A in order to M.
M Inumoi A	Moiki M ki A	In order to M, A.
A Inomoi M	Nunomoiki A ki M	*Nevertheless A although M.
M Inunomoi A	Nomoiki M ki A	Although M, A.
D Irau R	Nurauki D ki R	*Therefore D because of R.
R Inurau D	Rauki R ki D	Because of R, D.
D Inorau R	Nunorauki D ki R	*Nevertheless D although R.
R Inunorau D	Norauki R ki D	Although R, D.
C Isoa P	Nusoaki C ki P	*Therefore C given P.
P Inusoa C	Soaki P ki C	Given P, C.
C Inosoa P	Nunosoaki C ki P	*Nevertheless C although P.
P Inunosoa P	Nosoaki P ki C	Although P, C.

The *-ed entries are those for which no grammatical English translation seems possible.

Again we notice that in these logically sophisticated regions of the vocabulary, the fineness of grain of the Loglan lexicon far exceeds that found in any natural language. It is likely that Loglan speakers trained--as, for some time, all will be--in the coarser distinctions of the natural languages will find most of this domain of forethought causal connectives too fastidious to be usable. But it is also possible that the availability of these extremely precise logico-causal distinctions will lead some loglanists down linguistic pathways which will eventually take them to some powerful reflections not easily formulated in the natural languages, and that others will then follow in their linguistic footsteps in pursuit of these same or equally powerful new insights.

Lexeme KI: Infixes for Forethought Coonectives (Keks)

There are just two of these, **ki** and **kinoi**; and they are used as infixes with the prefix elements of the forethought connectives described in Lexeme KA.

Lexeme KIE: The Left-Parenthesis

KIE is monolexic, its sole member **kie** being the left or leading parenthesis. **Kie** is often represented in text by the punctuation mark [(], but is always pronounced [kyeh] when read aloud. Used with **kiu** below.

Lexeme KIU: The Right-Parenthesis

KIU is also monolexic, its one member **kiu** being often represented in text by the mark [)] but pronounced [kyoo] when read aloud. Used with **kie** above.

Lexeme LA: The Name Operator

LA has two allolexes, **la** and **laa**, the latter being a special operator for Linnaean names. **La** has the sole function of generating designations based on ordinary, non-Linnaean names. **La** may be used to precede either a sequence of one or more name-words (**La Djan Pol Djonz**) or a string of one or more predicate units (**La Redro Nu Herfa** = 'The Red-Haired One'). The initial letters of name-words are always capitalized; but the words in **la**-marked predicate strings are also given capital initials in text.

Names are seldom unique; but they are always used as unique designations in the contexts in which they appear. Thus, there will normally be only one person answering to the call 'John Paul Jones!' in the situation in which **La Djan Pol Djonz** is used, and only one red-haired person in the context in which **La Redro Nu Herfa** is successfully used.

To use names vocatively, the vocative operator HOI (q.v.) is used in place of **la** in either type of expression.

Laa is used only with Linnaean polynomials; see Sec. 2.13.

Lexeme LAE: Indirect Designation Operators

LAE has two members, **lae** and **sae**. **Lae** is used to designate something by operating on a sign or address of that something. Generally, **lae** precedes a designation of some linguistic entity, i.e., a quotation, although it need not; any object that can serve as a sign of some other object will do. But if it precedes a quotation, **lae** enables us to designate the referent-of-the-referent of that quotation. For example, (i) **Liegai, War and Peace, gai** designates the English expression 'War and Peace'. That English expression, however, is often a sign of--a kind of label or address of--a certain volume in many English-speaking persons' libraries, namely an English translation of that Russian novel known in English as 'War and Peace'. If **lae** were used to precede (i), the resulting Loglan expression (ii) **Laeliegai, War and Peace, gai,** would no longer designate this English expression, but the volume in my library, say, which has this English title printed on its back. So when I ask you (iii) **Eo kambei mi laeliegai, War and Peace, gai,** what I am asking you to bring me is not that English expression, but the volume whose address it is. Similarly, if I don't know the name of the person who lives at 123 Main street, but I would like you to take this copy of "War and Peace" to him or her (and we were speaking Loglan) I could say (iv) **Eo kambei laeliegai, 123 Main Street, gai laeliegai, War and Peace, gai,** which is equivalent to saying in exceedingly careful English 'Please bring to the person whose address is '123 Main Street' the object with the title 'War and Peace'.'

Sae allows us to perform the inverse of this indirect addressing maneuver. By prepositioning **sae** before any argument, we may use the resulting expression to designate the sign or signs of which the designatum of that argument is the referent. For example, **Sae levi bukcu** might be used to designate the title or titles of "this book" that I am holding in my hand, say in whatever languages it has been translated, or to designate its "addresses" conceived in some other way, say some particular library's shelving code for it. Thus **sae** undoes what **lae** does; and so both **sae lae** and **lae sae** do nothing.

Lexeme LE: Descriptors

Descriptors are words that make designations out of predicate expressions. LE is an open lexeme presently composed of a group of l-initial words, all of which have a 'the'-like quality. In the following lists, **preda** stands for any predicate expression used descriptively...a "descriptive predicate", in the terminology of the grammar. Here is the entire current list of simple LE-words as they might operate on any **preda**:-

le preda The one thing I mean, or each of the set of things I mean, which I believe appears or appear to you to be a preda, or to be predas. E.g., **Le mrenu pa fumna** = 'The man was a woman/the men were women'. (Explicit plurals are managed with quantifiers; see NI.)

lo preda The mass individual composed of all the instances of preda there are. E.g., **Lo cutri ga djipo lo clivi** = 'Water is important to life.'

lea preda Each of the set of all things which are predas. E.g., **Lea humni ga razdou** = 'All humans are rational (give reasons).'

loe preda The characteristic or normal individual which best exemplifies the predas in the present context. E.g., **Loe panzi ga fotli loe humni** = 'The typical chimpanzee is stronger than the typical human.'

lue preda The particular set of predas, or apparent predas, which I have in mind. E.g., **Lue monca gorla ga numcmalo** = 'Mountain gorillas are few in number.'

lua preda The set composed of all the predas there are. **Lua ficli ga mutce laldo** = 'Fishes are very old.'

Compound LE-words may be formed by postfixing to one of these simple LE-words either (a) a DA-word (any variable) or (b) a TAI-word (any letter variable), and/or (c) one or more PA-words. Usually the PA-words are spatials or temporals. The possessive adjectives **lemi** ('my'), **lemu** ('our'), **leda** ('X's') and **leTai** ('T's') are formed in this manner. So also are the demonstrative descriptors **levi** and **leva** ('this...' and 'that...') and the tensed descriptors **lefa, lena** and **lepa** ('the future...', 'the present...' and 'the former...') as in **Lefa bragai** = 'The future king'. Some of these LE + PA words are very long, e.g., **Lepacenoinacefa bragai je la Frans** = 'The-once-and-future king of France'; but all of them can be plainly deciphered. Thus, **le + pa + ce + noi + na + ce + fa** means nothing more mysterious than 'the-before-and-not-now-and-after (king of France)'. All these words behave grammatically just like **le**.

Lexeme LEPO: Event Operators

These are the LE + PO and LE + ZO (see PO and ZO) compounds that make arguments out of sentences. They are recognized as LEPO words by the preparser. LEPO words are known as event operators, although they designate properties (for example, with **lepu**) and quantities (with **lezo**) as well. Their grammar is quite different from that of LE-words, which they resemble morphologically, in that LEPO-words but not LE-words take whole sentences as operands. Sometimes it looks as if the operand of an event operator is nothing more than a predicate expression--e.g., **lepo sucmi** = 'the swim'--but this is because the operand is here the simplest of all sentences, a one-word imperative. The event of swimming could be fully specified: **lepo da sucmi de di** = 'X's swim to Y from W'. So unlike the operands of ordinary descriptors, the operands of event operators may be embellished without limit: **Lepo lemi brudi ji la Djan, pa kamla lepo fomtitci, e stolo mu ne nirne** = 'The time my brother John came to dinner and stayed with us for a year.'

Notice that the events, properties or quantities described with event operators may be particular (**lezo da pa sucmi** = 'the amount X swam') or general (**lopo sucmi** = 'swimming'), or general but constrained in various ways: e.g., **lovipo sucmi** = 'The here-swimming (the swimming here)' and **lenapo penso** = 'The now-thinking (the thinking now)'.

Loglan event descriptions are philosophically of some interest. Their designata are the objects of thought, for example; and so event descriptions accomplish what in the Indo-European languages is accomplished by so-called "indirect discourse". Thus, 'He thought that it was going to rain' is translated into Loglan with an event description: **Da pa jupni lepo ti fa crina** = 'X opined the event of this (place) being later rained-on.'

Lexeme LI: The Left Quotation Operator

LI is monolexic and is the opening mark for ordinary, or "weak" quotation. **Li** is used with the close-quote **lu** to quote grammatical Loglan, usually in strings of two or more words; see LIE and LIU for other varieties of quotation.

Lexeme LIE: The Strong Quotation Operator

LIE is also monolexic; it is the leading mark of "strong" quotation. That is, it is the mark by which non-Loglan strings or even strings of malformed Loglan may be unequivocally quoted. **Lie** is used with two instances of any arbitrarily chosen word X which are placed at each end of the string to be quoted. The boundary marker X is usually a letter-word (see TAI) and each instance of it must be separated from the quoted string by a pause: **lieX, <quoted stringS, X.** /.X/ must not of course appear adventitiously within the string being quoted.

Lie-quotation is most often used to quote foreign language strings. When it is, the upper-case Loglan letter-word for the Latin letter which is initial in the Loglan predicate for that particular foreign language is used conventionally as the boundary marker X: **LieSai, No hablo Español, Sai. Sai** is the sign of **spana**, the predicate meaning 'is Spanish/an element/expression of the Spanish language'.

Lexeme LIO: The Number Designator

LIO is the monolexic number designator. **Lio** is used before mathematical expressions to designate numbers. For example, **Lio ne natra numcu** = 'The number one is a natural number' and **Ti langa ta lio topifemei** = 'This is longer than that by 2.5 meters.'

Lexeme LIU: The Single-Word Quotation Operator

LIU is used to quote single words. It is monolexic and used only before single, well-formed Loglan words to designate them. For example, **Liu liu logla purda** = 'The word 'liu' is a Loglan word'. See LI and LIE for other varieties of quotation.

Lexeme LU: The Right Quotation Operator

LU is monolexic and is the closing mark for ordinary, or "weak" quotation. **Lu** is used with the open-quote **li** to quote grammatical Loglan, usually in strings of two or more words; see LIE and LIU for other varieties of quotation.

Lexemes M1* through M11*

M1 is a machine lexeme inserted by the preparser before any instance of A that is connecting linked arguments.

M2 is a machine lexeme inserted by the preparser before any instance of KA that is connecting linked arguments.

M3 is a machine lexeme inserted by the preparser before any instance of KA that is connecting predicates.

M4 is a machine lexeme inserted by the preparser before any NO that it is negating a modifier.

M5 is a machine lexeme inserted by the preparser before any A that is connecting sentence modifiers.

M6 is a machine lexeme inserted by the preparser before any A that is connecting argument modifiers.

M7 is a machine lexeme inserted by the preparser before any PA that is inflecting a predicate.

M8 is a machine lexeme inserted by the preparser before any NO that is negating a marked predicate, as shown by PO, ZO or M7.

M9 is a machine lexeme inserted by the preparser before any ACI that is connecting predicates.

M10 is a machine lexeme inserted by the preparser before any A that is connecting predicates.

M11 is a machine lexeme inserted by the preparser before any AGE that is connecting predicates.

Lexeme ME: The Predifying Operator

ME is the monolexic operator that turns any designation into a predicate ("predifies" it). Prefixed to the first word in that designation, **me** creates a predicate with a meaning, often vague, that can be associated with that designation. Some examples: **Ta metu cadre** = 'That's a you-ish dress'; **Plizo le mela Kraislr** = 'Use the Chrysler'; **Da mela Nuiork** = 'X is a New Yorker'; **De mele sitci** = 'Y is of the city (citified? cityish?)'.

Such usages create "false compounds", i.e., forms like **metu, mela** and **mele** which are recognized as not being members of single lexemes by the preparser, as true compounds always are, and so are decompounded. Thus, **metu** is lexed as the two-lexeme string ME DA before it goes to the parser; see DA.

Lexeme NI: Quantifiers

The quantifier lexeme NI has in principle a limitless number of allolexes, since it contains not only the digits **ni ne to te fo fe so se vo ve** ('none' 'one' 'two' 'three' 'four' 'five' 'six' 'seven' 'eight' 'nine'), but all the compounds of whatever length that might ever be made from them. Thus, **netotefofesosevove** as well as **neni** ('ten') is an allolex of NI.

NI also includes **ho,** the interrogative quantifier ('How many?' or 'How many of...?'); thus **Ho da pa kamla** = 'How many of X came?' It includes three of the non-numerical quantifiers as well, namely **re ro ru** ('most' 'many' 'enough'); the other two r-words, **ra ri** ('all' 'few'), having certain compounding duties to perform--they make the cardinal and ordinal compounds that will be classified as PREDA--must occupy a separate lexeme, namely RA.

Other allolexes of NI are **sa si su** ('approximately (all)' 'at most (one)' 'at least (one)'), which may either be prefixed to other NI-words (**savo** = 'approximately eight' **sivo** = 'at most eight' **suvo** = 'at least eight') or used alone with their default values understood (**sa** = 'nearly all', **si** = 'at most one', **su** = 'at least one'). Finally, there is a series of mathematical expression-building operators, not yet complete, consisting of the decimal point **pi** [.], the double-zero **ma** [00], the triple-zero **mo** [000], the division sign **kua** [/] (read 'over'), the inverse division sign **kue** (not available on this font, but read 'under'), the addition sign **poi** [+], the subtraction sign **nio** [-], the negation sign **niu** (not available), the multiplication sign **tia** [*], the power sign **sua** [|] (read 'to the ...-th'), the left and right parentheses **kie** and **kiu** [( )], which in this context, may often be read as 'the quantity', and for more complex nesting, the left and right brackets **gie** and **giu** [[ ]], also often readable as 'the quantity'. Mathematical expressions formed with these operators--for example, **tokuafo** [2/4] and **tosuaniufo** [2|-4] (read '2 to the power minus 4')--are also NI words.

NI has another group of allolexes which are formed of some numerical expression plus one or more letter-words of the TAI-Lexeme, e.g., **nevepinifomeimei** [19.04mm]. These are the dimensioned numbers; they, too, as well as the TAI words themselves, may be components of mathematical expressions. For example, consider **tobeisuato** [2b|2] = 'two-bee-squared' vs. **tobeikiusuato** [2b)|2] = 'two-bee-the-quantity-squared'. The "grammar" of such expressions, i.e., the set of compounding rules for NI compounds, has not yet been written.

A final type of NI compound may be formed by prefixing **ie** (see Lexeme IE) to any other NI word, simple or compound, e.g., **ieto** or **ietobeigiusuato.** Such quantifiers have the sense of 'which...of...?', as in **Ieto le mrenu** = 'Which two of the men?'...or, indeed, as in **Ietobeigiusuato le protoni** = 'Which two-bee-the-quantity-squared of the protons?' The internal structure of such compounds may be as complex as the speaker or writer likes; but grammatically, such expressions count as just another quantifier.

There is one usage convention involving NI and TAI words that resolves an ambiguity that cannot be solved be preparsing. This is the apparent ambiguity of the expression **nei preda.** Is (i) **nei** a quantifier, in which case this is an indefinite description meaning 'n predas'? Or is (ii) **nei** a letter-variable, in which case it means 'n is a preda'? Consider that the speaker has already established that the letter word **nei** [n] in some mathematical expression is to represent a certain large unknown number. May da then lift it out of that expression and use **nei** as a quantifier, saying, for example, ***Nei mrenu pa kamla** (interpretation (i))? Da may not; the parser would declare this expression to be unparsible (and so I have starred it). What is happening here is that **Nei** has been lexed as a TAI word (interpretation (ii), which happens to be the correct one); but so-lexed, the expression will not parse. Standing alone, as it does, **Nei** will not be parsed as a NI word, i.e., a quantifier. Whatever da's intent, da's auditor will hear **Nei** as a letter variable, presumably as a replacement of some longer argument in which 'n' figures; because that is grammatically what **Nei** is. So what da has said would translate into the equally ungrammatical English non-sentence '*Enn is a man came'. In other words, da would like to use **nei** as a quantifier, but may not because any well-behaved Loglan parser will treat it as an argument.

The way out of this dilemma is to adopt a certain usage convention. We may use the dimensioned number **nenei** [1n] whenever we would like to use the number represented by **nei** [n] as a quantifier. **Nei** is not a quantifier; but **nenei** [1n] is. Moreover, this usage is mathematically correct. The number one is the identity operator in this group; so 1X = X for any X. So it doesn't matter mathematically whether we say **nei** or **nenei**. But it does matter grammatically; for only by using **nenei** will the parser behave in the way that we intend. To English-speakers, this usage has a certain cost. It will be a trifle awkward, at first, to have to say **Nenei mrenu pa kamla** ('One-enn men came') when one feels, following English usage, that one ought to be able to say more economically ***Nei mrenu pa kamla.** But later on I trust we loglanists will discover the satisfaction of speaking precisely and so, understandably. Besides, by observing this convention we preserve the much more important use of TAI words as anaphoric variables; see TAI below.

Lexeme NO: The Negation Operator

There is just one negative in Loglan and it is used in all of the ways that the various negatives of the natural languages are used. **No** is very widely distributed in Loglan grammar, occurring in 13 rules. In particular, **no** occurs (i) in R28 where it is used before predicate words and other predicate units (**Da bilti no nirli ckela** = 'X is a beautiful non-girls' school'); this is the close-bound negative that we often express with 'non-' in English. It occurs (ii) in R34 where it is used to negate kekked predicate units (**Da bilti no ke botci, e nirli ckela** = 'X is a beautiful not both boys and girls' school', that is, a school for beautiful boys or beautiful girls or beautiful things that are neither boys nor girls, but not for both beautiful boys and beautiful girls; the scope of this **no** is confined to the kekked pair of predunits **ke botci, e nirli**). **No** is used (iii) in R62 before modifiers (**Titci no na** = 'Eat (but) not now') where the preparser will have marked it with a leading M4. It is used (iv) in R64 where it negates kekked modifiers (**Titci no ke vi, e na** = 'Eat (but) not both here and now'); the preparser will also have marked this **no** with M4. It is used (v) and (vi) in R134 and R143 to negate tensed or otherwise marked predicates (**Da no pa mutce gudbi takna** = 'X was not a very good talker'); this is long-scope **no**, the tense operator having spread the scope of **no** over the entire predicate expression. The preparser will have looked ahead and found the **pa** or other scope-extending mark on the other side of **no**, and then inserted M8 before the **no** to tell the parser that this is a long-scope negative of a predicate it is dealing with. **No** is also used (vii) in R149 to negate identity predicates (**Da no bi de** = 'X is not equal to Y'); and (viii) in R151 to negate kekked predicates (**Da no ke gudbi takna, ki bilti ckela** = 'X is not both a good talker and a beautiful school'; here the scope of **no** covers the entire kekked predicate). It is also used (ix) in R159 to negate statements (**No da bilti ckela** = 'It is not the case that X is a beautiful school'); as suggested, this **no** has the sense of the logician's 'It is not the case that...'. **No** is used (x) in R161 to negate kekked sentences (**No ke da bilti ckela ki de gudbi takna** = 'It is not the case that both X is a beautiful school and Y is a good talker'. It is used (xi) alone in R168 where **No** is a fragmentary utterance, perhaps an answer. **No** is used (xii) in R183, where it is used with a "gap" (a comma or a **gu**) before any utterance, including fragmentary ones (**No, na la Nemen** = 'No; in January'); in such uses **no** often has the sense of being one of several fragmentary answers being made in the same utterance. Finally, **no** occurs (xiii) in R184 where such leading negatives may be preceded by free modifiers (**Ia no, na la Nemen** = 'Certainly not; in January').

NO also occurs in /noV/ compounds (**noa, noe,** etc.) that are some but not all of the A words. In this use it is rather like NOI (see next lexeme) which also combines with vowels and other elemental lexemes to produce A and other kinds of connective words.

*Lexeme NOI: The Negation Suffix

This lexeme does not occur as a separate word. It is used only by the preparser to detect some sorts of A, CA, ICA, KA and KI words, namely those that end with the suffix **-noi.** For example, the A words **anoi** and **enoi,** the CA words **canoi** and **cenoi,** the I words **Icanoi** and **Icenoi,** the KA words **kanoi** and **kenoi,** and the KI word **kinoi** all require that NOI be recognized by the preparser, in particular, by that portion of it that analyzes compounds, in order that they be assigned to their proper lexemes.

Lexeme NU: Conversion Operators

The simple allolexes of this lexeme are **nu fu ju,** the three conversion operators which in Loglan effect the passive voice and a great many other "place-shifting" constructions as well. Thus, **nu** exchanges the meanings of the 1st and 2nd places of a predicate (**Da nu bloda de** = 'X is hit by Y'); **fu,** of the 1st and 3rd places (**Da fu ketpi de** = 'X is the departure-point from which ticket Y is valid'); and **ju,** of the 1st and 4th places (**Da ju ketpi de** = 'X is the carrier on which ticket Y is valid'). NU also includes compounds like **nufu** and **nufuju,** and the series of numerically marked conversion operators **nute** (= **fu**), **nufo** (= **ju**), **nufe, nuso,** etc., an alternative series of conversion operators that accommodate extremely long place-structures. All these words are recognized by the CPD-lexer.

NU words may also be used in incomplete utterances: **Da nu bloda** = 'X is hit', **Da fu ketpi** = 'X is a departure-point of some journey for which tickets are required', and **Da ju ketpi** = 'X is a carrier on which tickets are required.'

Lexeme PA: Inflectors/Adverbs/Prepositions

PA is a "portmanteau" lexeme in that its allolexes belong to 5 semantically distinct series: tense operators, location operators, modal operators, causal operators, and the predicate marker **ga.** These very different kinds of words are members of the same lexeme not because their meanings are similar but because their grammar, though not their usages, happens to be identical.

As the title of this lexeme suggests, any of the PA words may be used in three grammatical contexts: (i) as an inflector of some predicate (**Da pa titci ta** = 'X ate that'); (ii) as an adverbial modifier of the main predicate expression of the sentence (**Da titci ta pa** = 'X eats that earlier') or, when linked to an argument by a JI word (q.v.), as a local modifier of that immediately preceding argument (**Da titci ta ji pa** = 'X eats that earlier one, i.e., that one that was before'); and (iii) as the prepositional head of a phrase or a clause which modifies the main predicate expression (**Da titci ta pa la Ven** = 'X eats that before nine' or **Da titci ta lia lo horma** = 'X eats that like a horse') or, if linked to an argument, then as a local modifier of that argument (**Da titci ta ji lia lo horma** = 'X eats that (thing) that is like a horse').

If the prepositional meaning of a PA word is known, then its adverbial meaning can generally be inferred by regarding the adverb as an ellipsis of a phrase or clause headed by that same word used as a preposition or conjunction. Thus, we may infer that **Da titci ta vi** = 'X eats that here' is short for **Da titci ta vi ti** = 'X eats that in/at this (place)'; hence that the sentence without **ti** means 'X eats that here'. Similarly, the inflectional meaning of a PA word may usually be inferred by constructing a phrase using that word as a preposition and some designation of the place or time of speech as the object of that preposition. Thus **Da pa titci** may be regarded as shorthand for **Da pa ti titci** = 'X, before this occasion on which we are speaking, eats'; hence 'X ate'.

So much for the three contexts in which PA words may appear. We may now describe the five semantic types of PA words:-

Tense Operators: These are **pa na fa**, the simple past, present and future tenses, respectively, and their numerous derivatives and compounds. For example, there are the continuous tenses **pia nia fia** ('was ...-ing' 'is ...-ing' 'will be ...-ing', which yield 'since...' and 'ever since' for the prepositional and adverbial senses of **pia**, respectively; I shall let the reader work out the corresponding senses of **nia** and **fia**. There are the habitual tenses **pua nua fua** ('habitually ...-ed' 'habitually ...s' 'will habitually ...'). There are the intervalized tenses **pazi paza pazu** (the immediate, intermediate and remote past, for example) as well as **fazi faza fazu** and even **nazi naza nazu** with their analogous meanings. There are the nine narrative or compound tenses, the first element of each compound relating the point of speech to the reference point in the narration...often called the "narrative present", the second relating the reference point to the event. The first three of the narrative tenses are the so-called "perfect" tenses of concluded action **papa napa fapa** ('had ...-ed' 'has ...-ed' 'will have ...-ed'); the next three are the "progressive" tenses of predicted action **pafa nafa fafa** ('was going to...' 'is now going to...' 'will be going to...'); the last three are the less often used but equally interesting tenses of "coincident" action-- coincident because the reference point and the event coincide in these tenses-- **pana nana fana** ('was then ...-ing' 'is just now ...-ing' 'will be ...-ing'). Beyond the narrative tenses are the quantified tenses, of which **rana rona rena rina suna tona nena nina** ('always' 'frequently' 'usually' 'rarely' 'sometimes' 'twice' 'once' 'never') are just a few of the many quantified possibilities which the loglanist is invited to explore. Various combinations of the above are also possible, e.g., **papia** ('had been ...-ing') and **pazufazi** ('was long ago going to ... immediately'). There are the connected tense words such as **pacenoina** ('no longer' or 'once', literally 'before-and-not-now'). Finally, any PA word may, of course, be incorporated into a descriptor via the LE + PA compounding formula; for example, one of the narrative tenses **pafa** might be so incorporated, as in **Lepafa**

ditca = 'The-one-who-was-going-to-be a teacher'; see Lexeme LE. Some of these tensed descriptors are very long, but even the longest so far built are easy to decipher. Take **Lepacenoinacefa bragai.** The **pacenoinacefa**-part of this descriptor unravels easily as 'before-and-not-now- and-after'; whence the whole expression might be elegantly translated as 'The once-and-future king.')

Even this long list of PA words is not complete. Loglanists are invited to explore their many possibilities.

Location Operators: These are **vi va vu,** 'at' or 'in', 'nearby' and 'far away', and their many derivatives, e.g., **via vii viu,** and so on. This list is incomplete.

Modal Operators: Unlike the tense and location operators, which are inherently systematic, the modal operators are a disordered set. We can do no better than list the ones in current use; no doubt others will be added.

ciu	(ciktu)	as much as/to the same degree as...
dia	(dilri)	for/on behalf of...
biu	(bivdu)	in manner/mode.../by method...
kae	(kanse)	re/concerning/as for/with regard to...
lia	(clika)	like/as/in the way that...
lui	(pluci)	for/in order to please...
mou	(mordu)	as well as/in addition to...
hea	(helba)	with...'s help/through agent...
sea	(setfa)	instead of/in place of...
tie	(trime)	with..., a tool or means

Notice how the modals differ from the case tags of the DIO Lexeme. In a certain sense, modal phrases may be used to extend the place-structure of nearly any predicate; so they may be thought of as "itinerant cases". But because such phrases can turn up nearly anywhere, they may never be distinctive features of any predicate. Therefore, the modal preposition may never be omitted from its argument, as a case tag may be, relying on the auditor to infer the modality of that argument from its context. So the modality of an argument must always be explicitly announced by some attendant modal preposition.

Loglanists are just beginning to explore the uses of modals as inflectors and adjectives. We expect adventurous speakers to make many interesting discoveries in these uncharted waters. With a little thought, almost any of these strange new usages may be sensibly interpreted. For example, what does **Da durzo de hea** mean? In particular, what is the adverbial sense of **hea**? Well; if you think about it, **hea** used non-inflectionally and without an argument must at the very least must mean that X did Y with the help of someone else, that is, "helpedly". So **Da durzo de hea** must mean 'X does Y with help'. What about **hea** as an inflector? Reasoning analogically from, let us say, the sense of **pa** itself in these three positions--as a preposition **pa** means 'before (this designated time)', as an adverb 'before some undesignated time (presumably inferrable from context)', and as an inflector 'before this particular time, namely the point of speech'--**Da hea durzo de** must mean that X's doing Y was helped by someone or something current, perhaps a person present at the time of speech. Could we translate it as 'X does Y with your help', you the listener? Extending the currency principle to another case, what does **Da sea durzo de** mean? As a preposition **sea** means 'instead of'. So I would assume that the specimen means that X did Y instead of someone or something else, someone or something which was in some sense present at the time of speech. Could it mean that X does Y instead of your doing Y? I.e., in your place? But note that the speaker, too, is present at the point of speech...but perhaps less interestingly so, since da is always present. As I say, loglanists are just beginning to explore this vast new domain of meanings that has been opened up by the machine grammar work.

Causal Operators: These are the PA words which, when prefixed with /i/, form the causal connectives which are members of the I-Lexeme; q.v. Here the same sixteen causal relationships are invoked by causal prepositions; and these same sixteen words may, if interpreted by ellipsis, be used adverbially and even as inflectors. The most common use of these elemental causal operators is as prepositions, and that is the sense of the English translations given here. Some of these meanings do not exist in the natural languages, so their Loglan meanings are occasionally hard to think out.

kou C	because of cause C
nukou E	therefore/with effect E
nokou C	despite cause C
nunokou E	nevertheless (unexpected) effect E
moi M	because of motive M
numoi A	so action A
nomoi M	despite motive M
nunomoi A	nevertheless (unexpected) action A
rau R	because of reason R
nurau D	thus decision D
norau R	despite reason R
nunorau D	nevertheless (unjustified) decision D
soa P	because of premise(s) P
nusoa C	thus consequence(s) C
nosoa P	despite premise(s) P
nunosoa C	nevertheless (unentailed) consequence C

When the prepositional sense is plain, the adverbial and inflecting senses are easily inferred.

Ga: **Ga** is the fifth kind of PA word, and there is just one of its kind. **Ga** is a boundary-marker; it is used as the left boundary marker of any predicate whose left boundary would otherwise be unclear. **Ga** is therefore one of the "spoken punctuation marks" of Loglan as is suggested by its g-initial form. (The other boundary markers are **ge gi go goa goi gu gue**, q.v.) The left boundary of a predicate happens to coincide with the inflecting position of the PA words; so **ga** has the grammar of PA words, and is therefore lexemically a PA word even though it is semantically quite unlike the rest of them. Moreover, unlike other PA words **ga** is used only in the inflecting position, and then only when the immediately preceding word--exclusive of the "free modifiers" which, for these and other purposes, constitute a kind of "grammatical noise"; see Lexeme UI--is a predicate. Thus **ga** is used only in contexts like **Le sadji ga fumna** ('The wise one is a woman') from which its omission or removal would cause an unintended description to be heard: **Le sadji fumna** = 'The wise woman'.

It is considered bad usage to use **ga** where its removal would not cause this ambiguity. Thus ***La Selis, ga fumna** and ***Tu ga fumna**, although grammatically understanable, do not occur in well-formed Loglan speech. Here, then, is another grammatical superset, only some of whose members are actually used.

Lexeme PAUSE: The Pause-Comma

This is the lexemic element, represented by a pause in speech or a comma in writing, which has survived the preparser's inspection of its context and has been found to be genuinely lexemic, that is, necessary for the parser to be told about. The two written expressions of PAUSE are as a comma [,] and as a pound-sign [#]. Pauses come to the preparser as [,]s. The preparser then removes all instances of [,] which are not lexemic, i.e., which, like the morphemic pauses after name-words or before vowel-initial connectives, do not actually figure in the grammar rules, and rewrites the others as [#]s. Only [#]s occur in the preparsed strings which are delivered to the parser.

Lexeme PO: Abstraction Operators

There are exactly two of these, the event-abstractor **po**, and the property abstractor **pu.** There is a third abstractor, **zo**, which abstracts quantities; but because psuedo-instances of **zo** are occasionally generated while making acronymic PREDAs, the preparser, which must identify these kinds of PREDAs, requires that **zo** have its own lexeme; see ZO. This is a consequence of the fact that /z/ is the acronymic hyphen. So **CIO** is **CailzO**, which the preparser will read as a compound formed of the lexemes TAI + I + ZO and then classify it as a PREDA. Thus the preparser uses ZO in its lexing operations. If it weren't for this, **zo** would be a member of PO, which it is like in all grammatical respects.)

The two PO words and ZO occur in 3 contexts: (1) Close-bound or "short-scope" PO/ZO occurs just before a predicate word or predunit, and in this context the scope of the abstraction is just that single word or predunit; e.g., **Da po sucmi ditca** = 'X is an act-of-swimming teacher', presumably one who teaches that activity. It should not be inferred that abstraction is necessary here. This same claim can be made with the shorter concrete metaphor in **Da sucmi ditca** = 'X is a swim(mer) teacher', and loglanists generally prefer these concrete forms. Where precision is needed, however, the **Da po sucmi ditca** metaphor is available to the loglanist just as the gerund with '-ing' is available to the anglophone...and with perhaps less ambiguity about what is intended. (2) Long-scope PO/ZO occurs when it is followed by either (2a) a pause-comma or a **gu,** in which case the scope of abstraction is the entire ensuing predicate expression, including any arguments, e.g., **Da po, sucmi ditca lo frasa** = 'X is a case of being a swimmer teacher of French' (perhaps one who teaches French while swimming?), or (2b) either an argument or a modifier, in which case no other mark is needed to tell the parser that the PO has long scope; e.g., **Da po de sucmi ditca lo frasa** = 'X is a case of Y's being a swimming teacher of French'. In both these contexts, the operand of PO/ZO is a sentence, even though it may be a sentence without arguments--i.e., an unspecified imperative--in context (2a). (3) PO/ZO also occurs in LEPO compounds (q.v.); in this descriptive context the abstraction is always long-scope; e.g., **Lepo de**

sucmi ditca lo frasa = 'The event of Y's swim-teaching French'.

The semantic distinction between **po** and **pu**, between what we call in English "events" and what we call "properties", is perhaps the most difficult to understand in Loglan. The difference between these two operations is clearest in the descriptive context, that is, between the designata of event-descriptions and those of property-descriptions as abstracted, let us say, from the same predicated relationships. Any differences found here should apply, of course, to the other contexts. Let us examine, then, both an event-description and the property-description abstracted from the same relationship.

Lepo evidently abstracts a case, state, condition or event of any length--as long as an epoch or as short as a sneeze--from some predicated relationship. **Lepu**, in its turn, may abstract a property or quality from the same relationship. Suppose the predicated relationship is a motherhood between a mammalian mother X, an offspring Y and a father Z. We note first that the event-descriptions abstracted from such relationships are time-bound; like stories, mammalian motherhoods have beginnings, middles, and ends, and can apparently involve substantial segments of the "life-lines" of several or even many individuals. On a common understanding of motherhood, this one will involve those segments of X's and Y's lifetimes from the moment of her conception of Y and either her death or Y's, whichever is earlier. Of Z's lifetime, at least in many mammalian species, only a brief snippet need be considered to be part of X's motherhood, namely that snippet in which his sperm was delivered to her in some fashion. Of course more may be involved; in species in which pair-bonding takes place, more is involved. But this is sufficient to describe the event-state-condition of motherhood in mice, for example. We note that the thing that **lepo** designates about these 3 individuals is laid out in space-time. It has a beginning, a duration and an end. And the relevant segments of each of the three life-lines describe paths through space as well.

The property description of this same relationship, in contrast, is essentially time-free. It is the least set of facts which would establish the truth of that predicated relationship if it were true. That Y developed from an ovum produced by X's ovarian tissue is one of them; that that ovum was fertilized by a spermatazoan produced by Z's testicular tissue is another one. Are there any others? It would seem not. These two properties of these three individuals would, if known, be sufficient to establish X's maternity of Y by Z in any court of law...or, more likely, in any laboratory. And that, it would appear, is what a property is: it is that lean abstraction from the richness of the world that is just sufficient to enable us to know the truth of some matter.

The event-description is, in contrast, a "fat" concept. The designata of descriptions like **lepo da mamla matma de di** ('The event-state-condition of X's mammalian motherhood of Y by father Z') are often rich and complicated segments of the world; and they have a richer dimensionality than properties do. They have duration, for example, as properties do not. The designatum of **lepu da mamla matma de di** ('The property of X's mammalian maternity of Y by father Z') consists, in contrast, of two very simple but sometimes difficult to establish facts; and those facts are strangely free of time and other complications.

In sum, designata of **po**-abstractions are rich, multi-dimensional objects distinguished by duration; the designata of **pu**-abstractions are least sets of sufficient facts. The designata of **zo**-abstractions are the leanest of all, of course, being simply numbers...and sometimes uninteresting numbers at that. While the amount of heat in this room, and the amount of blue in that painting may well be interesting numbers, **lezo da mamla matma de di** is not. How shall we measure the quantity of mammalian motherhood, as it relates to three specified individuals? Give it one if it obtains between them, zero otherwise? Not a fruitful enterprise. If one or more of the three required participants remains undesignated, however, then the number **lezo da mamla matma**, the amount of mammalian motherhood in which X is involved, for example, might have some interest.

In short, the **zo**-abstraction is the least widely applicable of the three abstractions. **Zo** is useful only with those properties or relations which science has managed to quantify in some way. Thus, until we know how to measure the blueness of a painting or the motherness of an animal, **lezo ta blanu** and **lezo ta matma** will have designata we will not know how to find.

Lexeme PREDA: Predicate Words

Predicate words, while the most numerous items in any lexicon--they comprise about 80% of the Loglan dictionary at the present time--are called by only two grammar rules, R21-2. PREDA words enter the grammar through this narrow window in the rule group called "predicate units", and as predunits they are then variously elaborated as the noun-like structures of Loglan, the arguments, and the verb-like ones, the predicates. But because they all enter through the same narrow window, any predicate word may end up being either.

Morphologically, PREDA words come in five varieties:-

(1) The CC-bearing, V-final words of the lexicon like **preda** itself. These range from short borrowed words like **iglu** to long complex predicates like **rojmaosenmao** ('agronomist'). All PREDAs of this kind are identified by the lexer. They are by far the largest group of PREDA words that have actually been built so far; they include noun-like predicates like **mrenu** ('is a man'), verb-like ones like **godzi** ('goes from...to...by route...'), adjective-like ones like **corta** 'is shorter than...by amount...'), preposition-like ones like **bitsa** ('is between...and...'), and adverb-like ones like **mutce** ('is extreme in dimension...' which is more frequently used as an adverbial modifier of other predicates, as in **Da mutce sadji** = 'X is very wise'). Loglan makes no grammatical distinction whatever between these various kinds of predicate words...as is attested by their being members of the same lexeme.

(2) The numerical predicates made with the suffixes **-ri** and **-ra** which are members of the RA Lexeme; the exact compounding formula for these kinds of PREDAs is NI/RA (+ ...) + RA. This formula generates the two infinite sets of the ordinal and cardinal numbers, respectively, e.g., **neri** = 'is first in sequence...' and **nera** = 'is a monad', as well as certain irregular numericals like **sutori** ('is second or subsequent in sequence...') which are so much more efficient than the English circumlocutions which they translate that they have already crept into the English speech of some loglanists.

(3) The 6 predicate variables, which are two sorts: **bua bue buo buu**, the non-designating predicate variables used in logic (**Raba rabe rabua goi ko ba bua be ki be nu bua ba** = 'For every x, y, and predicate P, x P y if and only if y nu P x'); and **dui dua**, the demonstrative predicate variables, or "proverbs" as they are sometimes called in English grammar, which have meanings much like English 'do'. For example, 'He meant to go; and he did'. This sentence could be rendered into Loglan as **Da moi godzi, ice da dui.** Like all demonstratives, the **i**-final member of the pair is used for proximate

items, whether in time or space, the a-final one, for distal items.

(4) Acronymic PREDAs such as **CaiIzO** [CIO]; these may be made or borrowed at will by the user. Some Loglan acronyms, like **DNA** and **USA**, have simply been borrowed from existing acronyms in the natural languages...but of course they are then repronounced as Loglan words. Thus **DNA** in Loglan text is the abbreviation of the compound Loglan word **DaiNaiA**, and **USA** is short for **USaiA**. Acronymic predicates may also be derived internally, that is, from commonly used Loglan phrases whose frequency of use may have increased so much that the Zipf principle (that the length of a linguistic expression be inversely proportional to its frequency of use) will insist that a short Loglan expression be found for it. The increasing frequency of use of long, usually technical expressions (e.g., 'deoxyribonucleic acid') is, of course, the chief cause of acronym-formation in both the natural languages and in Loglan.

It is worth noting in passing that, while the part of speech of a compound letter-word, such as **MaiTai** [MT], is that of an acronymic PREDA, a single letter-word, such as **Mai** or **Tai**, is classified as an instance of Lexeme TAI, the letter-variables. Letter-variables, in turn, are one of the elemental forms of the Loglan argument. So a large grammatical shift takes place in the movement of **Mai** to **MaiTai**; see Lexeme TAI for further discussion of this point.

Finally, (5), there is the predicate interrogative **he**, which is of course also a member of the PREDA lexeme. **He** may be used in all places where any other PREDA may be used:- As a modifier: **Da he forli** = 'X is how strong?'; as a modificand: **Da mutce he** = 'X is very what?'; as the entire predicate expression: **Tu he** = 'You're what?' or 'How are you?; as the entire utterance: **He** = 'What?', often used in the sense of 'What did you say?' A more polite form of the last expression is **Eo he** = 'Please, what?'

In conclusion, the predicate words of Loglan are of an immense number and variety; but all of them have exactly the same grammar. To know how to use one of them is to know how to use them all.

*Lexeme RA: Numerical Predicate Suffixes

There are two of these, **ra ri**, and both are used both as ordinary NI words and to form the numerical predicates among the PREDAs. If it weren't for the use of Lexeme RA by the preparser, which must lex these numerical predicates by analyzing them as little word compounds, **ra ri** would, like their companions **re ro ru**, be allolexes of NI.

Lexeme TAI: Letter Variables

There are exactly 100 of these letter variables in Loglan, 52 of them are words for the Latin characters, 26 upper-, and 26 lower-case; and 48 of them are words for the Greek characters, 24 upper-, and 24 lower-case. In the sequel, the word 'letteral', which is an analog of 'numeral', will stand for the phrase 'letter character'. Thus, just as [1] is the numeral for the number-words 'one' and **ne**, so [t] is the letteral for the letter-words 'tee' and **tei**. For their construction, see Secs. 2.21-23.

All letter-words are members of the TAI Lexeme. TAI enters the grammar at just two points, in rules R79 and R81. In R79 TAI words enter as simple arguments, as in such utterances as **Tai mrenu** = 'Tee is a man'; in R81, they enter as the operands of the numerical descriptor **lio**, as in **Lio Tai numcu** = '(The number) Tee is a number', for this, too, is a possible use of a letter-variable. As arguments, the uses of TAI words are very similar to those of DA words, the replacing and personal variables, etc. Like DA words, TAI words may be used as possessives in LE + TAI compounds, e.g., **Letei kapma** = 't's hat'. Other compounds made with TAI are the dimensioned numbers formed by NI + TAI, e.g., **Sai corta Tai lio nemakeimei** = 'S is shorter than T by 100km (that is, by 100 kilometers)'. The word **nemakeimei** is, of course, a member of NI. A final use of TAI words in compounds is to make acronymic PREDAs; these are often strings of TAI or A words interspersed with NI words, the latter being confined to non-initial positions: e.g., **HaitoSaiOfo** [H2SO4]. Sometimes the morphological rules call for one or more instances of interstitial **-z-**, the acronymic hyphen, e.g., as in **CaiIzO**; see Sec. 2.29 for a fuller account of the construction of acronymic PREDAs.

Since simple TAI words are arguments and compound ones are predicates, this leads to a grammatical problem for the chemical abbreviations, some of which are simple TAI words, e.g., **Sai** or [S] for Sulphur, and others compound, e.g., **Caiza** or [Ca] for Calcium. This means that **Da Caiza** is a legitimate sentence (meaning 'X is Ca (i.e., Calcium)') but that **Da Sai** is not. In fact, **Da Sai** is a pair of arguments, perhaps an ordered couple, best translated 'X, S'. This lack of grammatical parallelism between the simple and compound letter-words is unfortunate, but it appears to be necessary and is easily accommodated. 'X is Sulphur' can be easily said symbolically in Loglan by the addition of another syllable: **Da meSai.** Thus the preidifer ME (q.v.) turns the argument **Sai** into a predicate form.

Assigning the simple TAI words to one lexeme and the compound ones to another is justified by the fact that doing so leads to two very considerable conveniences, surrendering either of which would make the language poorer. One convenience is that if simple TAI words are arguments, then a rich and virtually limitless mechanism of "anaphora" (replacement of long designations by short ones) is made available to the loglanist, with the result that even ordinary Loglan speech may very well become "mathematized": **Ama pa donsu Bai, Cai** = 'A gave B (gift) C', or **A pa donsu B, C** in the letteralized written form. We may expect that eventually such usages will exploit the full set of 100 Loglan letter variables. To give up this elegance by making simple TAI words predicates (whence ***Le Ama pa donsu le Bai le Cai**) would be to abandon this potentially powerful Whorfian experiment before it had been tried. So this convenience argues that at least the TAI words should be arguments.

A second and opposite convenience is assured if acronyms, which are nearly always compound letter-words, are treated grammatically as predicates. If this is done, then those acronyms which arise to fill the Zipfean need to shorten originally long predicate expressions--for example, as 'DNA' shortens 'deoxyribonucleic acid'--may be used in exactly the same ways as the longer expressions they replace. Thus, 'That's DNA' will replace 'That's deoxyribonucleic acid'; and in Loglan the same abbreviation will be **Ta DaiNaiA.** To make acronyms arguments, would be to undo this elegance. It would require that acronyms be prefixed by the predifier **me-** whenever their naturally predicative role was required...which would be very frequently. Thus, ***Da meDaiNaiA** would be the required form for the predicate 'is an instance of DNA'; and again a useful elegance would be lost.

The solution adopted preserves both elegances. But it requires that when acronyms are used in designations, they be treated as the operands of some descriptor. Sometimes **lo** is the best descriptor, as in **Donsu mi leva batpi je lo DaiNaiA** = 'Give me that bottle of DNA'. But sometimes **la** is best, as in **La USaiA groda gunti** = 'The USA is a big country'. It depends entirely on whether, in the given context, the designation so-generated is meant to be unique or not. The English description 'the USA', like its parent phrase 'the United States of America', is nearly always meant as a unique designation; therefore its translation into Loglan should probably be graced with **la**. Loglan makes such intentions explicit.

As mentioned, when it is desired to use a letter-word as a predicate, it must first be predified by **me**; see Lexeme ME.

Lexeme UI: Free Modifiers

These are the words which, unless initial in an utterance, are counted by the preparser as "grammatical noise". So UI words (along with several other items discussed at the end of this section) are removed by the preparser from the string to be parsed. They are restored to it by the postparser after the string has been parsed. We are justified in effecting this temporary removal of noisy elements from the string given to the parser because free modifiers are literally free to appear anywhere in any utterance, and so their appearance somewhere in a certain utterance gives no grammatical information about it. Free modifiers are therefore almost purely semantic devices. Their restoration after parsing assures that their semantic information will not be lost.

The entire grammatical significance of a UI word therefore lies in whether it appears at the head of its utterance or within it. When a UI word appears at the head of an utterance, it is taken to modify that utterance as a whole (**Ia mi ditca** = 'Certainly, I am a teacher'). When one appears non-initially, it is taken to modify the individual word it follows: **Mi ia ditca** = 'I, certainly, am a teacher (I don't know about the others)'; **Mi ditca ia** = 'I teach, certainly (I'm not sure what else I do)'. To follow a word with a free modifier is often to emphasize it. Thus the last two utterances might well be spoken /MIiaDITca/ and /miDITcaia/, respectively, while the first, with its free modifier in the Loglan-normal initial position, might receive no emphatic stress at all: /iamiDITca/.

UI words are of five semantic types: First, there are the **attitudinals**, which are used to express rather than report the speaker's attitude toward what da is saying, or toward some aspect of the world that is evoked by what da says. Second, there are the **discursives**. These are the words that call attention to how the units of the speaker's discourse are related to one another, or to the utterances of another speaker. Third, there are the **relative interrogatives**, the 'When?' 'Where?' 'How?' and Why?' series made by compounding any PA word with HU. Fourth, there are the **utterance ordinals** made by compounding any NI with FI. Finally, there are the **salutations**. These are the greetings, farewells, and expressions of gratitude and acknowledgement. All five kinds of UI words may be placed anywhere in an utterance, although unless special emphasis is intended, the initial position is stylistically preferred.

In general, UI words are extralogical; that is to say, they do not alter the truth-values of the sentences which they adorn. For a speaker to be convinced that da came, or to ask why da came, has no bearing on whether or not da came.

Now, in detail:-

Attitudinals. All attitudinals are VV-form words; all the VV-form words in Loglan are attitudinals except **ie**, the identity interrogative. The exceptional **ie** is more grammatically constrained than the other VV-form words but is related to them semantically, as the act of interrogating expresses an attitude; see Lexeme IE. The current list of simple attitudinals is:-

ae	Yes, I hope so/that/to...	
ai	Yes, I will/intend to...	(**-ai** is also L. u.c. cons. suff.)
ao	Yes, I want to...	(**-ao** is also Gk. u.c. cons. suff.)
au	I don't care whether...	
ea	Let's.../I suggest that...	
ei	Is that so?/Is it true that...?	(**-ei** is also L. l.c. cons. suff.)
eo	Please...	(**-eo** is also Gk. l.c. cons. suff.)
eu	I suppose/Let's suppose that...	
ia	Yes, that's certainly true/Certainly/I agree that...	
ii	Perhaps...	
io	Probably.../I think so/that...	
iu	I don't know whether...	
oa	I/you must...	
oe	I/you should...	
oi	I/you may...	
ou	It doesn't matter ethically whether I/you...	
ua	(completion) There!/At last!	
ue	(surprise) Well!/Oh!	
ui	(pleasure) Gladly/Happily...	
uo	(anger/annoyance) What!	
uu	(sorrow/regret) Alas.../Sorry that...	

Note that several VV-forms are still unassigned: **aa ee oo.** These may yet be given meaning as loglanists learn how to handle these disyllables which to anglophones still seem odd.

To expand the domain of attitudinal expression, any number of compound attitudinals may yet be constructed. For example, **uiua** has a clear meaning: pleasure + completion = satisfaction; **ueui** also has a clear interpretation...and a usage; for example, it might be used to greet a long-lost friend standing at one's door. But few of these attitudinal compounds have been made. Again, the loglanist is invited to explore a new semantical domain.

Discursives. All the discursive UI words are CVV in form; but by no means all the CVV-form words are members of the UI Lexeme. For example, all of the case-tags (DIO), most of the letter variables (TAI), and many of the modifying prepositions (PA) are also of CVV-form. So there is some potential for confusing CVV-forms that seem to have similar functions. The way to distinguish the discursive UI words from the prepositional PA and DIO words is to ask whether the word you're uncertain about may be used prepositionally. If it may be, it's probably either a PA word or a DIO word. If not, it's almost certainly one of the discursives.

But what about the many words with vaguely "adverbial" meanings? If the word in question changes the claim of the predicate of the sentence in which it occurs, it's probably a PA word; a further test is whether you can turn it into a preposition. But if the word in question seems to relate the utterance as a whole to some other utterance, stated or implied, it's probably a discursive.

Here is the current list of discursive UI words. Note that none of them may be used prepositionally, that the meanings are generally adverbial but that each such adverbial meaning seems to appeal either explicitly (**cia**) or implicitly (**coa**) to another utterance or, like **feu**, to allude to outside information:-

bea	(bleka)	For example/By way of illustration (cf. **gea**)
buo	(bufpo)	However/In contrast/On the other hand
cea	(cenja)	Changing topics/(New paragraph)
cia	(clika)	Similarly/Like the foregoing
coa	(corta)	Summarizing/In short
dau	(dakli)	Hence it is probable that
dou	(donsu)	Given/By hypothesis/Per assumption
fae	(fanve)	And vice versa (reverses the order of terms in a previous claim to form a new one)
fau	(fando)	Finally/In conclusion
feu	(fekto)	In fact/Actually/According to the facts
gea	(genza)	In particular/As an instance (cf. **bea**)
kii	(kliri)	Clearly/Obviously/Plainly/Of course
kuu	(kumtu)	Generally/Generalizing from the above
leu	(clesi)	Anyway/Anyhow/In any case/event
nie	(snire)	In detail/Closely examined
saa	(sapla)	Loosely/Roughly/Simply speaking
sui	(sumji)	Also/Too/Moreover/Furthermore/In addition
voi	(volti)	Skipping details/Without going into details

Notice that, like the DIO words, when a natural derivation of the discursive from its primitive mnemonic yields a TAI word, then the final vowel of that first derivative is changed to /u/. **Dau** and **leu** illustrate the /u/-transformation in the above list. What this means is that TAI words have derivational precedence over other CVV-form words.

Relative Interrogatives. These are the compounds made from any PA word plus the argument interrogative HU ('Who?'). The meanings of such compounds are always readily decipherable, since they are nothing more than contractions of prepositional phrases in which the object of the preposition, its operand, is always **hu**. So **nahu** means 'At what time?' or 'When?'; **vihu** means 'In or at what place?' or 'Where?'; **biuhu** means 'In what manner?' which is one of the many varieties of 'How?' (nearly any modal operator will fit here); and **kouhu** means 'Because of what?' or one of the numerous Loglan 'Why?'s. All the PA words--and there are many hundreds of them--are capable of generating such question words, and with much greater precision than is apparently ever available in the interrogative forms of natural language.

The surprising thing about these interrogatives is their very simple grammar. Like the true-false interrogative **ei**, they can turn any utterance into a question. The whole utterance is its operand. Thus **ei** turns **Toi tradu** ('This is true') into the question **Ei toi tradu** = 'Is this true?'. Similarly, **kouhu** turns it into the causal question **Kouhu toi tradu** = 'Why (because of what cause) is this true?' And three other varieties of 'Why?', namely **Moihu Rauhu** and **Soahu**, are available to the loglanist to lend surgical precision to da's inquiries.

Utterance Ordinals. This is the series of compounds formed by attaching the suffix **-fi** to any NI or TAI word. The results, **Nefi Tofi Tefi** and finally **Rafi**, allow the loglanist to number da's utterances--or, for that matter, to number any sequence of clauses or terms within an utterancce--with effects like English 'Firstly' 'Secondly' 'Thirdly' and 'Finally' or 'Lastly'. Letter-ordinals may also be employed: **Amafi Baifi Caifi** and so on; and for the purposes of outlining, the lower-case Latin and the two Greek series of utterance letter-ordinals may also be used: **asifi beifi ceifi, Amofi Baofi Caofi** and **afifi beofi ceofi**, etc. These words, too, are grammatical noise.

Salutations. There are four such currently assigned UI words. They are **loi** ('Hello') **loa** ('Goodbye') **sia** ('Thank you') and **siu** ('You're welcome'). Like the attitudinals the salutions are expressions of the speaker's feelings or desires, but this time, toward da's interlocutor or someone da wishes to speak with. No doubt there will be more such words in time. Their grammar is identical to that of the free modifiers. They may be used anywhere; or, if no special internal emphasis is desired, they are usually placed, like the other UI words, at the head of the utterance: **Loi mi bi la Djan** = 'Hello! I'm John'.

A Note on Other Free Modifiers. In addition to the UI words, there are three other structures that are handled as free modifiers in Loglan grammar. These are (1) the vocative marker HOI (q.v.) together with any string of one or more name-words that may optionally follow it (**Hoi nenkaa** = 'Hey, come in! vs. **Hoi Djan, nenkaa** = 'O John, come in!'); (2) any unmarked string of one or more name-words, that is, unmarked by either LA or HOI (**Djan Pol Djonz, nenkaa** = 'John Paul Jones, come in!'); and (3) the parenthetic marker KIE together with the parenthesized utterance and the closing parenthesis KIU which will always follow KIE in well-formed speech (**Kie Rafi kiu Djan, pa nenkaa** = '(Finally) John came in').

These three elements together with Lexeme UI constitute the complete list of free modifiers in Loglan.

Lexeme ZE: The Joining Operator

ZE is monolexic; it contains only **ze**, the "joining" or "mixing" operator which has the sense of 'and, jointly'. **Ze** is not a logical connective; it does not allow a number of connected claims to be made simultaneously. A claim made with **ze** is always a single claim. Thus, **Da redro, e cmalo** makes two claims about X, namely that it is red, and that it is small. **De redro ze nigro** makes only one claim about Y, namely that it is red-and-black mixed together. Perhaps it has red stripes alternating with black ones; perhaps it has red dots on a black field. But if a thing is **redro ze nigro**, it is not true that it is either red or black separately.

ZE may be used to join arguments or, as above, predicates. The designata of **ze**-joined arguments are like teams. If **Da ze de pa berti leva felstaga** ('X and Y, jointly, carried that log (fallen trunk)' then what is being asserted is that the team of them did. We may gather that it is most unlikely that either of them could have carried it separately.

The grammar of ZE is very similar to that of the afterthought connectives; see Lexeme A. However, since **ze**-binding is "tighter" than shek-binding, that is, since strings like **Da, e de ze di** = 'X, and Y and Z, jointly' will always parse as **(da e (de ze**

di)), grammatically "earlier" rules are required to effect **ze**-joints than shek-joints. This is what requires **ze** to be in its own lexeme.

ZE is also used by the preparser to recognize acronymic PREDA's. This is because pseudo-**ze**'s are sometimes generated by the acronymic hyphen **-z-**.

*Lexeme ZI: Magnitude Suffixes

There are three of these words, **zi za zu**, indicating small, intermediate, and large magnitudes, respecitvely, in their role as suffixes in tense (**fazi** = 'right away') and location (**vizu** = 'in this region') compounds; see PA. A separate lexeme is required to enable the preparser to recognize such compounds and also to identify some acronymic PREDA's, because, just as with ZE (q.v.), pseudo-instances of **za zi zu** may be generated in these words by the acronymic hyphen **-z-**.

*Lexeme ZO: The Quantity Abstractor

ZO is monolexic; it's sole member is **zo**, the quantitative abstraction operator. It has a grammar parallel to that of PO, q.v. The only reason **zo** occupies a separate lexeme is because, just as with ZE and ZI words, pseudo-**zo** syllables may be generated by the use of hyphen **-z-** in some acronymic PREDA's. If it weren't for this mechanical use of ZO by the preparser, **zo** would be a member of PO.

CHAPTER 4

GRAMMAR (UTTERANCE FORMS)

4.1 Design Objectives: The objectives that have controlled the design of Loglan grammar have been, first, to accommodate the rich variety of claims and designations found in natural language but, second, to do so with a grammar modeled on the predicate calculus, thus facilitating both the logical manipulation of its utterances by its users and the design of a language with parsimonious metaphysical assumptions. Third, it was to make the operations of that grammar natural enough to fit the human central nervous system so as to make the language speakable, and fourth, it was to be flexible enough to imitate the natural word orders of the planet's major tongues when desired...for example, in translation. But it was also, fifth, to be expressible in a rule set small enough to be easily learned by adults, possibly even under experimental conditions. Finally, sixth, the grammar was to be syntactically unambiguous for two reasons. The most obvious one was to make interaction with machines possible. Somewhat less obvious is a reason that is the same reason logicians and mathematicians require unambiguous codes, namely to make it possible rigorously to entertain implausible ideas...a prime requirement of a logical language.

Syntactic unambiguity was achieved heuristically, and therefore doubtably, as early as 1963, and was not to be formally demonstrated until 1982. A few years before that achievement a constructive-proof algorithm for demonstrating conflict-freeness in certain classes of computer languages had become available and was soon adapted to disambiguating human grammars. Syntactic unambiguity was lost again in 1984 when The Institute's small computer proved inadequate to the task of servicing the growing grammar. It was recently redemonstrated for the enlarged language when The Institute acquired more capacious computing machinery in 1986. During the Winter and Spring of 1987 a large backlog of planned new grammatical features were installed in the language one by one; and I was gratified to discover that the condition of conflictfreeness was easily returned to each time. In short, given our present tools for disambiguating human grammars, there is no reason to believe that Loglan grammar will not remain syntactically unambiguous for the indefinite future.

4.2 Definitions and Conventions: Let us first look at a typical grammar rule and provide a terminology for discussing its parts:

150:	kekpred	=> M3 KA predicate KI predicate	**Da ke ckano ki briga** = X **is both kind and brave.**
151:		=> NO kekpred	De **no ke ckano ki bunbo** = Y **is not both kind and a fool.**
	Local.		

This is the grammar rule by which "kekked", or forethoughtfully connected, predicates are formed. The specimens on the right illustrate each line of the grammar rule. The number on the left gives its position in the grammar. The remark 'Local' tells us that the rules which use 'kekpred' are all nearby.

Each numbered line in a grammar rule is called a **rule.** The sign [=>] in each rule may be read 'may be developed as' or 'may produce' and is called the **production sign.** The expression to the left of the production sign in a rule is called its **left-half.** If a rule has no explicit left-half, it is assumed to have the same left-half as the first preceding rule that has one.

Rules 150-1 are all the rules in this grammar that have 'kekpred' as their left-halves. A **grammar rule** is the complete set of rules in a given grammar that have a given left-half. Within a grammar rule, the order of rules is arbitrary.

This is a **context-free grammar.** In such grammars the left-halves of all grammar rules are single **elements. Right-halves** may either be single elements, in which case the rule is called a **replacement**, or they may be strings of two or more elements (up to about five in number), in which case, it's an **expansion.**

Elements which are written entirely in upper-case letters, like 'KA' and 'NO', or in a mixture of upper-case letters and numerals, like 'M3', are called lexemes. Recall that lexemes are sets of grammatically interchangeable words, roughly corresponding to the "parts-of speech" of conventional grammar.

Elements whose names are written at least partly in lower-case letters--for example, 'kekpred' and 'predicate'--are known as **gramemes.** By definition, each grameme in a grammar appears in the left-half of exactly one of its grammar rules. At present, there are 88 gramemes in Loglan grammar; so there are 88 grammar rules.

The remark 'Local' appended to the 'kekpred' grammar rule tells us that the rules that use this grameme are all nearby (actually, in the same "grammar group"; see below). Sometimes a series of one or more numerals occupies the space occupied here by 'Local'; these are references to the non-local rules which use the grameme defined by that grammar rule. The forward references are given first; the backward references, if any, are separated from the forward ones by a semicolon.

The right-halves of a grammar rule are called the **allograms** of its grameme. 'kekpred' has two allograms. When either gramemes or their allograms are referred to in text, their names will be shown in angle brackets. Thus we can say that <kekpred> has the allograms <M3 KA predicate KI predicate> and <NO kekpred>. Alternatively, we can write the Kekpred Grammar Rule as <kekpred => NO kekpred | M3 KA predicate KI predicate>. In this second formulation the bar [|] is used to separate the names of allograms and is read 'or'.

A grameme that appears in the allograms of another grameme, as <predicate> appears in one of the allograms of <kekpred>, is said to be **used by** that other grameme. A grameme which is used in one of its own allograms is said to be **recursive.** In general, we will find it advantageous to develop early in the grammar the gramemes that will be used by later gramemes.

A sequence of grammar rules which has been ordered by the above principle, and which terminates in a widely used structure, is called a **grammar group**, or sometimes just a **group.** Ideally, all the gramemes except the final one in a group are used only within that group. When this is true of a grameme in a group, that grameme is said to be a **local** to that group and so is marked 'Local' as above. Groups are usually given the name of the major non-local grameme with which they terminate. For example, <kekpred> is part of the Predicates Group (Rules 128-54) and is local to that group. The last grameme developed in the Predicates Group is <predicate>.

As shown in the example, each rule will be illustrated by a **specimen** of Loglan that it helped produce. When we need to refer to a rule or specimen, we will use the ordinal of the rule but prefix it with 'R' or 'S'; thus S150-1 are the specimens of R150-1. The part of a specimen that is in boldface corresponds to the part produced by its rule. In the specimens given for R150-1 above, all but the Loglan words [Da] and [De] were produced by the two <kekpred> rules; so all but those parts of the two specimens are shown in boldface. A corresponding pattern of bolding is shown in the English translations of the Loglan specimens.

Some rules use M- or machine lexemes. For example R150 uses M3. Machine lexemes are inaudible to the human ear, but, as we will see later, they are indispensable for machine parsing. They are in fact lexemes in **machine Loglan,** which may be thought of as that "dialect" of Loglan that machines will be able to read and hear. By convention, the name of a machine lexeme is formed of the letter 'M' followed by a numeral. There are at present 11 machine lexemes in Loglan: M1-M11. All are involved in one way or another with extending the limited 1-element **lookahead** of the **LR1 parser.** This Yacc-generated parser is one of the three components of our **machine grammar.** Another component is the **preparser** which, among other **mechanizing** functions, inserts machine lexemes into the strings to be parsed. These machine lexemes, together with certain other lexemes (like ERROR in the first grammar group), are of concern only to the machine and so are not part of **human Loglan.** These and other computational features are removed from the parsed string by the third component of the machine grammar, the **postparser.** It is the function of the postparser to **humanize** the parse by making it intelligible to humans. We retain these machine-oriented features in our exposition of the grammar in the current notebook only because some of our readers may be interested in how machines may be said to "understand Loglan". We assume that most readers, however, will wish to ignore all grammatical embellishments that have been put there solely for machines. So we will glance only briefly at these computational devices. Readers interested primarily in the computability of the language must go to other sources for detailed information about how this has been accomplished; e.g., **Notebook 1,** 1982.

4.3 The Structure of Loglan Grammar: The grammar rules of Loglan may be conveniently divided into twelve functional groups. We will discuss those groups here in the order in which the listener is likely to make use of them (the so-called "top-down" order). In that order they are the groups of rules that govern the formation of (1) Optional Punctuators, (2) Linked Arguments, (3) Predicate Units, (4) Descriptive Predicates, (5) Sentence Predicates, (6) Modifiers, (7) Argument Modifiers, (8) Arguments, (9) Term Sets, (10) Predicates, (11) Sentences, and (12) Utterances.

In general, later structures on the above list involve earlier ones. Thus, looking at the grammar from the "bottom-up", we would find that

Utterances require Sentences;
Sentences require Modifiers, Predicates and Arguments;
Predicates require Sentence Predicates and Term Sets;
Term Sets require Arguments and Modifiers;
Arguments require Descriptive Predicates and Argument Modifiers;
Argument Modifiers require Modifiers;
Modifiers require Arguments;
Sentence Predicates require Predicate Units and Descriptive Predicates;
Descriptive Predicates require Predicate Units;
Predicate Units require Linked Arguments;
Linked Arguments require Arguments again, which are thus the most "circular" structures in the language, in that they are used by nearly everything which they in turn use; and that

Optional Punctuators are used by nearly every other grammar group.

Therefore we will start with Group A, the Optional Punctuators, for these are at the "top" of the parse tree. We will then work "down" to Group K, which develops Utterances, for these are near its "root". The grammar is thus presented in its "top-down" order; it goes from the "leaves" (the lexemes) toward the "root" (the <utterance> grameme itself) of the parse tree. This is the same order, by the way, in which the rules appear in the formal (machine) grammar which was given to Yacc to build the LR1 parser. While this may not be the best order in which to teach grammar rules to human learners, it is the most economical one in that it minimizes forward references. It is thus most suitable for a work whose most enduring value to its users will probably be its utility as a reference work.

Group A. The Optional Punctuators
Rules 1-7

The three "optional punctuators" **gu, gue** and PAUSE, the latter being represented by a pause in speech and a comma [,] in text, are optional only in the sense that they may be omitted when they are not necessary. But in fact, usage requires that these punctuators be omitted when the intended grouping is clear without them; so they are seldom optional in any real sense. Oddly enough, the computer sometimes treats these punctuators as "present in spirit" even when they are omitted. S1 is an example of this.

1: err	=> ERROR Local.	Mu titci fa (**ERROR**) = We eat later. (The machine sees this as an error because it expects a **gu**. But if it supplies the dummy lexeme ERROR at the end of this utterance, it can complete the parse. We arrange for it to do this by making <err> an allogram of <gu> in R2, and of <gue> in R4.)
2: gu	=> err	Mu titci vi (**ERROR**) = We eat here.
3:	=> GU 121-2.	Mu titci vi **gu** le supta = We eat here the soup. (Without **gu** the utterance would be heard as 'We eat in the soup'; so **gu** is necessary here.)

Both **Mu vi titci le supta** = 'We here eat the soup' and **Mu titci le supta vi** = 'We eat the soup here' avoid the need for punctuation. The unpunctuated word orders tend to be stylistically preferred in Loglan but are certainly not obligatory. In fact, the judicious use of punctuation makes almost any conceivable word-order possible in Loglan, a feature which is very useful in translating whenever one wishes to reproduce the flavor of a natural language text as closely as possible.

There is also an invisible '(ERROR)' at the end of S3 as of nearly all specimens. But we don't show it here because the <err> allogram of <gu> is not part of R3.

4: gue	=> err	Da bilti ge cmalo nirli ckela **(ERROR)** = X is beautiful for a small-girls school. (Punctuation at the end of an utterance may always be omitted. Again, the computer will compensate for such "errors" by inserting the dummy ERROR Lexeme.)
5:	=> GUE 22-3.	Da bilti ge cmalo nirli **gue** ckela = X is a beautiful small-girls [pause] school, i.e., a school for small girls who are beautiful. (Here the **gue** serves as a right parenthesis matched with **ge.**)
6: gap	=> gu	Mu titci fa **gu** le mitro = We eat later the meat. (Again, an explicit mark is called for; to omit it is to generate the phrase **fa le mitro** = 'after the meat'.)
7:	=> PAUSE	Mu titci fa, le mitro = We eat later, the meat. (When <gap> is invoked, the required mark may be a comma in writing or a pause in speech.)

10, 15-6, 24, 59-60, 83, 87, 90, 92, 94, 98, 100, 130-1, 183-4.

As may be seen from the references, **gu** alone is required in only two rules, R121-2; **gue** is used in similarly few places, R22-3; but the triple option presented by <gap>, which may be executed by a **gu**, pause/comma or frequently by nothing at all, is extremely widely used. In fact <gap> is the most widely-used grameme in Loglan grammar. It provides an opportunity to mark the right boundary of phrases or clauses whenever such a boundary would be unclear without it. Loglanists tend to use pauses and commas at such points when dealing with human readers and interlocutors in conditions of low noise, and to use **gu**'s in high noise or when addressing computers or other unforgiving auditors. See the numerous rules listed above for the occasions on which these punctuators may be used.

Optional punctuators are not the only "punctuation marks" in Loglan. There are also six punctuators which are essential to the structures they mark and so may never be omitted. These are the "grouping operator" **ge**, which may be found in R22-3 along with the optional **gue**; the "inversion operator" **go**, found in R48; **goa**, which marks the V-O-S word-order in R155; and the "fronting operators" **gi** and **goi**, found in R165-66. In addition, there is a punctuator-like member of the PA-Lexeme, **ga**, which is used exclusively for marking the left boundaries of otherwise unmarked predicate expressions when these would be absorbed by just-preceding descriptions; see R139. For example, **ga** in **Le mrenu ga sadji** = 'The man is wise' prevents **Le mrenu sadji** = 'The man (sort of) wise one' from being heard. <Gap> would work here--for example, ***Le mrenu, sadji** also parses in the required way--but is regarded as bad usage (and so is *-ed) because there are some kinds of descriptive arguments after which <gap> fails to produce the required separation. **Ga** always works and so is preferred.

This is the entire punctuation system of Loglan. See the rules cited for the details.

Group B. Linked Arguments
Rules 8-19

Je and **jue** are the two preposition-like words that attach strings of one or more arguments to predicate words. Sometimes the predicate word involved is buried in a predicate expression (**Da kukra je lo litla, grobou** = 'It's a faster-than-light ship'); sometimes it is the last word in a description (**Le selrispe farfu je la Djek** = 'The proud father of Jack'). In either case, the linking words have the effect of binding a string of elements into a single unit. Thus if the je and **jue** of S8 were removed, the string remaining would be composed of three distinct arguments: **To ketpi da de** = 'Two tickets, X, Y'. The two linking operators bind this triad into a single argument: 'Two tickets to destination X from point-of-departure Y'. The present group of rules shows how to construct the right part of such expressions, the "linked arguments". The predicate units, like **ketpi**, to which they are attached are constructed in the next group; see R26.

8:	juelink	=> JUE argument Local.	To ketpi je da **jue de** = Two tickets to X **from Y.**
9:	links1	=> juelink	**Jue de = From Y.**
10:		=> juelink links1 gap Local.	**Jue de jue di = From Y on W.**
11:	links	=> links1	**Jue de = From Y.**
12:		=> links M1 A links1	**Jue de, a jue di = From Y or from W.** (The pause after de is a "morphemic pause", necessary for the resolution of a.)
13:		=> M2 KA links KI links1 174.	**Ke jue de ki jue di = Both from Y and from W.**
14:	jelink	=> JE argument Local.	To ketpi **je da** jue de = Two tickets **to X** from Y.
15:	linkargs1	=> jelink gap	**Je da = To X.**
16:		=> jelink links gap Local.	**Je da jue de = To X from Y.**
17:	linkargs	=> linkargs1	**Je da = To X.**
18:		=> linkargs M1 A linkargs1	**Je da, e je de = To X and to Y.**
19:		=> M2 KA linkargs KI linkargs1 26, 173.	**Ke je da ki je de = Both to X and to Y.**

When links are connected, as they are for example in S18, the linking words need not be repeated. Thus, Loglan **Je da, e de** conveys the same notion as S18 does just as English 'To X and Y' conveys the same notion as 'To X and to Y'. Sometimes in both languages, however, this degree of explicitness is desired.

Links and linkargs are used in only three non-local rules. Links are used in R172, and linkargs in R26 and R173. In R172-3 both links and linkargs appear as fragmentary utterances, such as might be answers to **Jue hu** and **Je hu** ('To/from/by whom?')

questions. Linked arguments enter the main stream of the grammar at just one place, however, namely in R26 of the next group, where they are attached to predicate words.

Notice that two machine lexemes have been introduced in Group B. They are M1 which marks the "eks" (the **a**-form afterthought connectives) which connect both links and linkargs; and M2 which marks the "keks" (the **ka**-form forethought connectives) which connect these objects. The reader will recall that machine lexemes are put in place by the preparser, one of whose tasks is to extend the limited 1-element lookahead of the machine. Human brains do not need these warnings since our capacity to remember and inspect long strings is so much greater than that of any parsing algorithm used currently by machines.

Group C. Predicate Units
Rules 20-33

Predicate units, or "predunits" as we will sometimes call them, are either single predicate words, with or without certain inflecting operators such as **nu**, **no** or **po**; or they are strings of such possibly inflected predicate words which have been made into a predicate unit by **ge**, or by a **ge** matched with a subsequent **gue** (R22-3); or they are arguments that have been turned into predunits by the "predification" operator **me** (R24); or they are predicate words which have been augmented by the attachment of one or more linked arguments (R26).

20:	predunit1	=> PREDA	Ba **sucmi** = Something **swims.**
21:		=> NU PREDA	Ba **nu sucmi** = Something **is swum to.**
22:		=> GE despredE gue	Da briga **ge musmu janto** = X is brave **for a mouse hunter.** (The <gue> is not activated here; it would be redundant if it were.)

Ge is the grouping operator; it is only meaningful if it precedes a string of two or more predunits. Thus ***briga ge musmu,** while grammatical, is proscribed as bad usage, for it can mean nothing other than what **briga musmu** ('brave mouse') already means. Again, redundant marks are avoided. <despredE>, which is the operand of **ge**, is made in the next group; see R45-6.

23:		=> NU GE despredE gue	Da **nu ge briga janto** = X **is a quarry of of brave hunters.** (**Nu** is one of the conversion operators; and **ge** extends the scope of **nu** over the entire <despredE> string, in this case **briga janto,** which then takes its place-structure from its last term, in this case **janto.** Thus X is a kind of quarry.)
24:		=> ME argument gap Local.	Ba **mela Ford** = Something **is a Ford.** (**Me** turns any argument into a predicate unit with a meaning associated with that argument.)

25:	predunit2	=> predunit1	Da **kukra** grobou = X is a **fast** ship.
26:		=> predunit1 linkargs	Da **kukra je lo litla,** grobou = X is a **faster-than-light** ship.
	Local.		

R26 is the primary use of the linked arguments made in the preceding group. Their use in descriptions also passes through this rule. Usage imposes a certain restraint on the use of R26, however. The <predunit1> to which linkargs is attached by it must not, by convention, be the last unit in a string unless that string is a descriptive predicate, i.e., one used in making arguments. Thus **Le mutce kukra je lo litla** = 'The thing which is very much faster than light' is an acceptable use of linked arguments. In fact the link word is necessary if this description is to be heard as one argument rather than two. The link word is also necessary in **Da kukra je lo litla, grobou** in which the argument is linked to a non-final predicate unit in a string. (We will call this the "internal specification" of a predicate.) But arguments are not allowed to be linked to final units in predicate strings when these are being used as sentence predicates. Thus while ***Da mutce kukra je lo litla** (presumably to render 'X is very much faster than light') is perfectly grammatical, it is a proscribed usage because there exists a preposition-free form that does the same job more elegantly: **Da mutce kukra lo litla.** Again, usage in Loglan avoids redundant markings, such as this quite unnecessary **je.** The same link words are not of course redundant inside predicate strings, or to link a descriptive argument to other arguments. But they are redundant if used to link a sentence predicate to its argument set, or the arguments in that set to one another.

This is the first instance we have encountered of the "grammatical superset" phenomenon. By this I mean that we have written a rule of grammar that generates a domain of grammatical utterances which is larger than the domain of "good utterances" allowed by usage. Another way of saying this is that not all grammatical utterances are acceptable as good usage. Any economically written set of grammar rules will occasionally generate such effects. They are perfectly harmless. We need only add a set of usage rules to such a grammar to ensure that any grammatical utterance that meets the usage rules will also be interpretable. We have at present no way of interpreting the difference between ***Da kukra je de** and **Da kukra de;** so we do not use the former. There are other supersets in Loglan grammar; I will call the reader's attention to them as we encounter them.

27:	predunit3	=> predunit2	Da **sadja** = X **is wise.**
28:		=> NO predunit3	Da **no bunbo** = X **is no fool.** (This is short-scope negation; it applies to a single predunit. Long-scope negation is accomplished elsewhere.)
	Local.		
29:	predunit4	=> predunit3	Ti **nigro** = This is black
30:		=> predunit4 ZE predunit3	Ti **nigro ze blabi** = This **is black-and-white (mixed).** (**Ze** is a special connective which has the sense of mixing properties.)
	Local.		
31:	predunit	=> predunit4	Ti **blanu** = This **is blue.**
32:		=> PO predunit4	Ti **po blanu** = This **is a state of being blue.** (This is short-scope abstraction; other uses of **po** take whole

			sentences as operands; see R130-1.)
33:		=> ZO predunit4 34, 36, 49-50.	Ti **zo blanu** = This **is an amount by which something is blue.**

Zo is kept out of the PO-Lexeme only because the preparser needs to recognize quasi-ZO in lexing acronyms. This is because **-z-** is the acronymic hyphen, and when it precedes /o/, as it does in **CaiIzO** [CIO], the result looks like a compound of TAI + I + ZO to the preparser. If **zo** were part of PO, it could not lex this acronym correctly. For this entirely mechanical reason--which probably has no meaning for human lexers--PO and ZO must be kept lexemically separate.

Predunits are the building blocks with which "descriptive predicates" (predicates used as "nouns") and "sentence predicates" (predicates used as "verbs") are made. These will be constructed in the next two groups. No machine lexemes have been necessary in the construction of predicate units.

Group D. Descriptive Predicates Rules 34-48

These are the predicate expressions which, when used as operands of any of the descriptive operators--**le** and kin, for example, or the name operator **la**, or any quantifier--create that special kind of argument known as nouns or noun-phrases in Indo-European grammar. We call such arguments "descriptions" because they employ a predicate expression to "describe" some feature of the designated thing.

The following sequence of rules differs from that in the next group, in which sentence predicates are made, primarily in one feature: descriptive predicates may have kekked head units, e.g., the **ke forli ki sadji** part of S35; sentence predicates may not. It turns out that ambiguities are generated if sentence predicates are permitted to have kekked head units, while the same constructions are quite unambiguous in descriptions. So kekked head units are provided in this rule group--the <kekpredunit> of R35 and 37-8--and absent from the next.

34:	despredA	=> predunit	Le **forli** = The **strong one.**
35:		=> kekpredunit	Le **ke forli ki sadji** = The **both strong and wise one.**
36:		=> predunit CI despredA Local.	Le **denro simba ci janto** = The **dangerous lion-hunter.** (Without **ci** the predunits in the string would group left, thus 'The dangerous-lion hunter'.)
37:	kekpredunit	=> M3 KA descpred KI despredA	Le **ke forli ki sadji** = The one who is **both strong and wise.**
38:		=> NO kekpredunit Local.	Le **no ke forli ki sadji** = The one who is **not both strong and wise.**

Another machine lexeme is introduced here, M3 which extends the parser's lookahead over KA again, this time to announce to the parser that a predicate is being kekked. Unmarked keks connect arguments.

39: despredB	=> despredA		Le **sadji** = The **wise one.**
40:	=> CUI despredC CA despredB 52, 54.		Le mutce **cui fizdi forli ce sadji** = The one who is very **physically-strong and** (very) **wise.**

Cui is the left boundary mark of left connectands formed of more than one predunit. Without **cui** the single units on either side of the "shek" (the CA-connective) are taken as its connectands. No similar mark for right connectands is necessary.

41: despredC	=> despredB	Le **mrenu** = The **man.**
42:	=> despredC despredB 52.	Le **mutce bunbo mrenu** = The **very foolish man.**

This is the rule by which predicate strings are generated. Because it is left-recursive, all unmarked predicate strings are left-grouping.

43: despredD	=> despredB	Le **sadji** = The **wise one.**
44:	=> despredD CA despredB 56.	Le **forli canoi sadji** = The **strong if wise one.**
45: despredE	=> despredD	Le **mrenu** = The **man.**
46:	=> despredE despredD 22-3.	Le **forli canoi sadji mrenu** = The **strong if wise man.**
47: descpred	=> despredE	Le **troku hasfa** = The **stone house.**
48:	=> despredE GO descpred 58, 72-4, 76, 97.	Le **hasfa go troku** = The **house of stone.**

Notice that several grammar rules in this group besides the last one are non-local. All these other non-local rules are employed at similar points in the next group, where sentence predicates are made. But as all descriptive predicates may have kekked head units, care is taken to ensure that these structures borrowed from the descriptive sequence do not end up as the heads of sentence predicates; for there they would generate ambiguities.

The descriptive predicate grameme itself, <descpred>, is used as the right part of R58, the final rule in the construction of sentence predicates, and at various points (R72-4, 76 and 97) in the construction of arguments, which are made in Group G.

Group E. Sentence Predicates
Rules 49-58

This group of grammar rules is structurally parallel to the preceding one, the only difference being the one already noted, namely that sentence predicates may not have kekked head units. Note that some "despred" gramemes figure in these "senpred" rules, but that they are never initial in an allogram; this renders them harmless. Thus **mutce ke briga ki ckano** is a permissible sentence predicate while **ke briga ki ckano**

mrenu, with its kekked head unit, is not. The reason for this proscription is not hard to find. If **ke briga ki ckano mrenu** were permitted as a sentence predicate, as in **Da ke briga ki ckano mrenu**, presumably to mean 'X is both a brave and a kind man', then the parser could not distinguish between this use of keks to connect predicate units--it is **briga** and **ckano** that are being connected here and not **briga** and **ckano mrenu**--and the later use of the same keks in Group I to connect whole predicate expressions. This would generate an ambiguity between the two rules, for they would both be capable of generating the same string. So the auditor could not tell whether the speaker meant his keks to connect just preddunits, as above, or whole predicate strings, as in Group I. The latter possibility would give an entirely different parse tree, one with the interpretation 'X is both a brave person and a kind man'. As things stand, the second interpretation is the only legitimate one...precisely because kekked head units are not allowed in sentence predicates.

49:	senpred1	=> predunit	Da **mrenu** = X **is a man.**
50:		=> predunit CI senpred1 Local.	Da **simba ci janto** mrenu = X **is a lion-hunting** man.
51:	senpred2	=> senpred1	Da **sadji** = X **is wise.**
52:		=> CUI despredC CA despredB Local.	Da grada **cui simba janto ce sadji** = X is a great **lion-hunter and** (a great) **wise one.**
53:	senpred3	=> senpred2	Da **prano** = X **runs.**
54:		=> senpred3 CA despredB Local.	Da **prano a fleti** = X **runs and/or flies.**
55:	senpred4	=> senpred3	Da **mrenu** = X **is a man.**
56:		=> senpred4 despredD Local.	Da **simba mrenu** = X **is a lion man.**
57:	sentpred	=> senpred4	Da **simba fumna** = X **is a lion woman.**
58:		=> senpred4 GO descpred 128.	Da **fumna go simba** = X **is a woman of lions.**

All the gramemes in this tiny group are local except the last one, and that one, <sentpred>, figures in just one later rule: R128 in Group I, which makes the objects called simply "predicates". It is in this later group that the sentence predicates made here will be supplied with their "termsets": the strings of modifiers and/or arguments that complete them as predicates. Before that can be done the termsets must be constructed. The modifiers and arguments of which termsets are composed will be prepared in the next two groups.

The making of sentence predicates has involved no machine lexemes.

Group F. Modifiers
Rules 59-67

Setting aside the "free modifiers" of R176-9, which are quite differently distributed and composed, the (unfree) sentence modifiers which occur in Loglan utterances are of two sorts. They are either (1) phrases, such as **pa da** ('Before X'), in which some PA-word functions as a preposition and some argument, in this case a very simple one, functions as its "object" or operand; or they are (2) argument-less PA-words, such as **pa** itself, which function in the sentence like English "adverbs". Thus **pa** without an argument is an ellipsis for 'Before some assumed time, say the moment of speech' and thus means simply 'Earlier'. Group F is exclusively concerned with the construction of these two kinds of modifiers. Its first grammar rule, <mod1>, gives the basic architecture of Loglan modifiers, which is exceedingly simple. Notice that both the **pa** and the **pa da** varieties of modifiers may be optionally followed by the <gap> triplet, the choice of **gu**, pause/comma or nothing at all, which is one of the optional punctuators developed in Group A.

59:	mod1	=> PA gap	Mu titci **pa** = We eat **earlier.**
60:		=> PA argument gap	Mu titci **pa la Ven** = We eat **before nine.**
		Local.	(When a modifier comes at the end of an utterance an explicit gap would be redundant and so is not used.)

61:	mod	=> mod1	**Fa = Afterwords.** (Later/then/etc.)
62:		=> M4 NO mod	**No fa = Not afterwards.**
		Local.	

The parser needs to know what kind of **no** this is; so the preparser has looked ahead of this **no** (as the parser can't) and found that it is negating a modifier. So the preparser erects the signpost M4 before the **no** to inform the parser of its discovery. (The preparser is thus a kind of forward scout for the machine's parser...which snuffles along blindly--but very mentally--with its nose to the ground.)

63:	kekmod	=> KA modifier KI mod	**Ke fa ki pa da = Both afterwards and before X.**
64:		=> M4 NO kekmod	**No ke fa ki pa da = Not both afterwards and before X.**
		Local.	

65:	modifier	=> mod	**Pa da = Before X.**
66:		=> kekmod	**Kanoi fa ki pa = If later then earlier.**
67:		=> modifier M5 A mod	**Fa noa na = Later only if now.**
		89, 118.	

Two machine lexemes are introduced in this short group, M4 which marks those **no**'s that negate modifiers, and M5 which marks the eks that connect them. The keks of R63 do not need to be marked.

Group G. Arguments
Rules 68-116

This is the longest rule group in the grammar, comprising nearly a quarter of all its rules. But the argument rules nevertheless constitute a coherent group, with few gramemes used elsewhere until we get to <argument> itself. Once developed, arguments are then used very widely throughout the grammar...for example, in the preceding group to make modifying phrases.

The question may well arise: Why should not arguments precede modifiers in their development? The answer is that arguments use modifiers in their development as well, and that arguments unlike modifiers have a great variety of backward references. They are, recall, used as early as Group B, Linked Arguments. It seemed best to put them in a position where all they will use has already been built; and then let them have all the backward references they require. In short, arguments are the most circular structures in Loglan grammar in that they are more frequently embedded in themselves.

An argument is a designation, and in Loglan as in the natural languages there are many kinds of designations. A preview of what is to be found in this large and important rule group may therefore be in order. We will commence with "names" (R68-9, 80), and move on through "definite descriptions" (R72-6, 83) and the various kinds of "quotations" (R84-6). Then we will provide for "event descriptions"--called "indirect quotations" in European grammar--whose handling in Loglan (see R87) expresses one of the fundamental philosophical orientations of the language (namely that objects of thought and imagination are best treated as a subset of definite descriptions). Then we provide for the "indefinite descriptions" (**Ne mrenu** vs. **Le mrenu**) in R97-100 and 102; and finally the various ways of connecting arguments are given toward the end of the group; R103-16.

68: name	=> DJAN	La **Djan**, ditca = **John** is a teacher.
69:	=> name DJAN	La **Djan Djonz**, ditca = **John Jones** is a teacher.
179.		

The commas in S68-9 represent "morphemic pauses", that is, the pauses required after names which permit their resolution. But this is a morphological matter, and the grammar pays no attention to such pauses. In fact the preparser eliminates all morphemic pauses before presenting a string to the parser. The morphemic pauses between the parts of a multiple name are necessary in careful speech--otherwise the lexer would hear them as single names, e.g., **Djandjonz**--but such internal pauses are not represented by commas in text. It would distract the eye to do so.

70: mex	=> RA	**Ra** da ditca = **All of** them are teachers.
71:	=> NI	**Ne** da ditca = **One** of them is a teacher.
170.		

The <mex> grammar rule will be much more elaborate once the mathematicians set to work on it. But I have thought it best to keep <mex> rudimentary until the rest of the grammar is functioning smoothly.

72: descriptn	=> LE descpred	**Le ditca** pa gudbi = **The teacher** was good.
73:	=> LE mex descpred	**Le te ditca** = **The three teachers.**

74:	=> LE arg1 descpred	**Le le ditca gu bukcu = The teacher's book.**
75:	=> LE mex arg1	**Le to da** ditca = **The two of them** teach.
76:	=> LA descpred Local.	**La Ditca** ga sadji = **Teacher** is wise.

Note the "possessive" provided in R74. An alternative possessive form will be provided in R88 in which the word order will be possessor-possession, as in the Romance languages: 'El libro de la maestra'. Like English ('The teacher's book' and 'The book of the teacher') Loglan has both possessive orders. As usual our policy is to provide ways of speaking every common natural word-order and let usage decide between them.

Note also, from R76, that descriptions may be used as names. When this is done the predicate words in the **la**-marked description are all capitalized in text, and the expression has, like other names, the sense of being a unique designation in that context. Thus there is only one **La Farfu** in the Smith family, just as there is only one Rat in **The Wind in the Willows.** Names may not be used as predicates, however. Thus ***Da Djan** cannot be used to mean 'X is a John' because this expression is not parsed as a sentence. (It is *-ed because it lacks a pause-comma before the name.) The **Djan** in **Da, Djan** as parsed by our grammer is neither a name nor a predicate but one of the free modifiers. It is a vocative, in fact, which modifies **Da.** So the proper rendering of **Da, Djan** into English is 'X, John' as if one were answering a question asked by John about the identity of something. The proper rendering of English 'X is a John' into Loglan is **Da mela Djan.** In this expression the predifier **me** turns **la Djan** into a predicate (see R24); and **Da** is the first argument, or "subject", of that predicate.

77: arg1	=> DA	**Da** ditca = **X** teaches (is a teacher).
78:	=> HU	**Hu** ditca = **Who** teaches?
79:	=> TAI	**Tai** ditca = **T** teaches.
80:	=> LA name	**La Tam,** bi ba = **Tom** is (that) someone.
81:	=> LIO TAI	**Lio nei** cmalo = **The number n** is small.
82:	=> LIO mex	**Lio te** cmalo **lio nei** = **The number three** is smaller than **the number n.**
83:	=> descriptn gap	**Le ditca** ga gudbi = **The teacher** is good.
84:	=> LI (utterance LU)	**Li, Le ditca ga gudbi, lu** steti = **Quote Le ditci ga gudbi close-quote,** is a sentence. (The commas here are stylistic only, i.e., neither morphemic nor lexemic.)
85:	=> LIU (word)	**Liu lu** purda = **The word lu** is a word.
86:	=> LIE (X, string, X)	**Lie Dai, Deutschland, Dai** dotca purda = **The string 'Deutschland'** is a German word. (These commas are obligatory. 'X' is any Loglan word that does not appear in the quoted string. I usually use the initial letter of the Loglan word for the language in which string is written, in this case D for **dotca.**)
87:	=> LEPO sentence gap	**Lepo da pa cluva de,** viekli = **That X**

		Local.	**loved Y** is clear.

The parenthetic portion of R84 is carried invisibly through the first parsing of the utterance and then parsed "re-entrantly" as an independent utterance. The parenthetic portions of R85-6 are not even seen by the parser. It is assumed that they either do not need to be (R85) or cannot be (R86) parsed.

88:	argmod1	=> JI argument	Da **ji la Djan,** ditca = The X **who is John,** teaches. (**Ji** identifies.)
89:		=> JI modifier	Da **ja vi le hasfa** ga kicmu = X, **who is in the house,** is a doctor. (**Ja** predicates.)
90:		=> JIO sentence gap Local.	Da **jio brudi de,** murpeu = The X **who is a brother of Y,** is a seaman (seaperson). (**Jio** identifies.)

Pe is also an allolex of JI and, like the 'de' of Romance-style possessives, puts the possessor last. Thus the phrase **pe le ditca** in **Le bukcu pe le ditca** ('El libro de la maestra') is also a production of R88.

91:	argmod	=> argmod1	Da **jia sucmi** = X, **who (incidentally) swims.** (**Jia** predicates.)
92:		=> argmod M6 A argmod1 gap Local.	Da **jia sucmi, e jia prano** = X, **who (incidentally) swims and runs.**
93:	arg2	=> arg1	**Da** = **X.**
94:		=> arg1 argmod gap Local.	**Da ji de** = **The X who is Y.**
95:	arg3	=> arg2	**Sai** = **S.**
96:		=> mex arg2 Local.	**Te Sai** = **Three of the S's.**
97:	indef1	=> mex descpred Local.	**Te solda** pa kamla = **Three soldiers** came.
98:	indef2	=> indef1 gap Local.	**Te solda** ga ditca = **Three soldiers** are teachers.

Either the comma or the **gu** of <gap> can be used here in place of **ga** to separate the first argument from its predicate. In fact ***Te solda, ditca** parses in substantially the same way as **Te solda ga ditca** does. But I have starred it because it is probably not a good idea to adopt this usage, despite its tempting economy. The reason is that if the first argument ends in a linked argument, <gap> will not successfully separate it from an upcoming predicate; instead <gap> will allow the intended predicate to be absorbed into its first argument. For example, **Le farfu je le botci, talna** does not say that the father of the boy is an Italian, as a careless speaker might have intended; instead it is a designation of some boy-fathering Italian: 'The father-of-the-boy (type of) Italian'. Replacing the comma with **ga** conveys the intended meaning unequivocally: **Le farfu je**

le botci ga talna = 'The father of the boy is Italian'. Thus the stronger marker **ga** is necessary in some cases. Since **ga** will work in all cases and <gap> only in some, it seems wiser for loglanists to habituate themselves to the always-successful marking move...even though **ga** is less economical than the pause/comma permitted by <gap> that would often do the job.

The frowned-upon uses of <gap> form another unused portion of a superset of grammatical possibilities.

99: indefinite	=> indef2	**Te solda** = **Three soldiers.**
100: Local.	=> indef2 argmod gap	**Te solda jia nigro** = **Three soldiers, who are (incidentally) black.**
101: arg4	=> arg3	**Bai** = **B.**
102:	=> indefinite	**Ne fumna** = **A (one) woman.**
103:	=> arg4 ZE arg3	**Bai ze Cai** = **B and C jointly.**
104: Local.	=> arg4 ZE indefinite	**Bai ze ne fumna** = **B and a woman jointly.** (**Ze** among arguments has the effect of forging single, team-like entities.)
105: arg5	=> arg4	**Bai** = **B.**
106:	=> KA argument KI arg6 Local.	**Kanoi Bai ki Cai** = **If B then C.**
107: arg6	=> arg5	bei groda **cei** = **b** is bigger than **c.**
108:	=> DIO arg6	**Mau cei** gi bei groda = **Than c,** b is bigger. (**Mau** is one of the optional case-tags of the DIO-Lexeme. Derived from **cmalo** it marks the lessers in "greater-lesser than" relationships. For the uses of the fronting operator **gi,** see R165-6.)
109:	=> IE arg6	Mau **ie cei** gi bei groda = Than **which c** is b bigger? (DIO-words are always omissible; **Ie cei gi bei groda** means the same thing. Yet **mau** is not redundant. It suggests in advance the kind of predicate that is coming up.)
110:	=> LAE arg6 Local.	Donsu da **lae bei** = Give X to **whoever has address b** (or of whom b is a sign in some other sense).
111: arg7	=> arg6	bei groda **cei** = **b** is bigger than **c.**
112:	=> arg6 ACI arg7 Local.	bei groda cei, e **dei, onoici fei** = b is bigger than c, and than **d or f but not both.** (**Onoi** is "exclusive or" and **-ci** turns its connectands into the right connectand of **e,** i.e., it hyphenates them.)

113:	arg8	=> arg7	**bei** groda **cei** = **b** is bigger than **c.**
114:		=> arg8 A arg7	bei groda **cei, e dei, onoi fei** = b is bigger than **c and d, or f but not all three.** (Without **-ci** these "afterthought" connectives group left.)
	Local.		
115:	argument	=> arg8	**bei** groda **cei** = **b** is bigger than c.
116:		=> arg8 AGE argument	bei groda **cei, ege dei, onoi fei** = b is bigger than **c, and d or f but not both.** (The **-ge** suffix groups all that follows it to the right, and thus has the same effect as a **-ci** suffix on the 2nd connective; cf. S112.)
	117; 60, 24, 14, 8.		

Notice that <argument> has only one forward reference, and that is the first rule in the next group. Most references to <argument> are backward, which only means that arguments have already been widely used. Indeed, we may recall that they have been used to construct one class of modifiers, namely prepositional phrases (R60); they have been used to build one kind of predunits, namely those composed of arguments preceded by the predifier **me** (R24); and of course arguments figured in linked arguments (R8 and 14).

The single forward reference to R117 leads of course to the most common use of arguments in Loglan utterances, their contribution to the "terms" and "termsets" which are major constituents of nearly every Loglan utterance.

Group H. Terms & Term Sets
Rules 117-127

A term is either a modifier or an argument. So terms are strings of one or more arguments and/or modifiers in any order. Term sets are strings of none or more terms and include the connections of such strings with one another. Group H is really two groups, both of them very small. One of them, R117-20, constructs <terms>; the other, R121-7, constructs <termset>. The uses of these two gramemes are as follows:-

Terms are used as the first arguments ("subjects") of sentences, and include any immediately following modifiers. Thus the argument-modifier pair **La Djan, na la Fomen** in **La Djan, na la Fomen, traci ti** = 'John, in May, travels here' is an instance of <terms>. But so also are the strings of fronted modifiers and/or arguments which come before the subject of a sentence. The only difference is that terms that come before a subject must be set off from it by **gi.** For example, **Ti na la Fomen** in **Ti na la Fomen, gi da traci** = 'Here in May, X travels' is also an instance of <terms>. **Gi** is the fronting operator; it announces that a string of such fronted terms has been concluded.

Termsets, in contrast, are always deployed after their predicates. For example, **ti** in **Da traci ti** is a <termset>. Indeed, all sets of "sutori" (second-and-subsequent) arguments, including any sentence modifiers with which they may be mixed (**ti fa** in **Da traci ti fa** = 'X travels here later'), or any strings of modifiers, or of arguments and modifiers mixed together, when they follow their predicates--for example, as **fa, na la Fomen** in **Da traci fa, na la Fomen** ('X travels later, in May') follows **traci**--are instances of <termset>.

As we will see, termsets are made of terms. But there are two differences in the resulting structures. First, termsets, but not terms, may be null, i.e., represented by the right boundary marker **gu** or by nothing at all; see R121. Second, termsets (but again not terms) may be connected to each other. Just why these two maneuvers are useful can probably only be discovered by using them. But I can give you a preliminary glimpse of their utility for termsets by pointing out that they (and not terms) are grammatically attached to sentence predicates; see R128. These predicate-containing constructions then, together with their possibly null termsets, become the "predicates" of Loglan sentences; and such predicates may be connected to each other in every possible way...including ways which allow them to have "joint termsets". For example, in the English sentence 'John loved and hated Mary', 'Mary' is probably intended by the speaker to be a joint termset of the connected pair of predicates 'both loved and hated', that is, to be the "direct object" of both "verbs" (to use an older grammatical terminology). Of course we cannot be sure of this in English; but in Loglan we can be. It is just this kind of potentially ambiguous construction that in Loglan necessitates either nullifying or truncating individual termsets in order to make room for joint ones. For example, the two possible renderings of 'John loved and hated Mary' into Loglan are (i) **La Djan, pa cluva, e tsodi la Meris,** in which **la Meris** is an individual termset, and (ii) **La Djan, pa cluva, e tsodi gu la Meris,** in which it is a joint one. What has happened in (ii) is that in it e and **gu** have nullified the individual termsets of **cluva** and **tsodi**, respectively, and made room for a joint one.

Such maneuvers are never required of terms. But if this structural distinction seems too intricate for easy recall, you may prefer to remember a simpler, positional one: terms occur ahead of their predicates; termsets occur after them. This surface property will take you a long way toward managing their uses properly.

117:	term	=> argument	Da pa cluva **de** = X loved **Y.**
118:		=> modifier Local.	Da pa cluva **fa** = X loved **later.**

119:	terms	=> term	Da pa cluva **de** = X loved **Y.**
120:		=> terms term 155, 157-8, 165-66, 171.	Da pa cluva **de fa la Ven** = X loved **Y after nine.** (The order of terms is syntactically unimportant. Thus **Da pa cluva fa la Ven, de** means approximately the same thing.)

Notice that all these references are forward. In effect, they list the many uses of <terms>. R155 deploys it as the delayed subject of "goa sentences" (sentences in V-O-S word order); R157 uses it as the fronted modifiers of a goa-sentence; R158 uses it as the subject, accompanied by any following modifiers, of a declarative sentence in Loglan-normal (S-V-O) word order; R165-6 arranges for its use as the fronted arguments, with accompanying modifiers, of sentences in O-S-V word order, but also as the fronted modifiers or prenex quantifiers of any utterance; and in R171 <terms> accommodates fragmentary utterances, for example, answers to **Hu, Vihu** and **Nahu** ('Who?', 'Where?' and 'When?') questions.

We now use <terms> to develop <termset>.

121:	termset1	=> gu	Da pa cluva de, e tsodi **gu** fa la Ven = X loved Y and hates (someone), **(both)** after nine. (**Gu** represents the null termset and turns **fa la Ven** into a joint termset. A comma is insufficient here; therefore this allogram is not <gap> but <gu>.)
122:		=> terms gu Local.	Da pa cluva de fa la Ven, e tsodi **di gu** va do = X loved Y after nine, and hates **W, (both)** near Q. (Here **gu** truncates the 2nd termset and prepares for a joint termset, **va do,** a modifier of both predicates, to be spoken.)

Notice that it is <gu> and not <gap> that is used to truncate or nullify an individual termset. This is because the use of a comma or a pause at such points can lead to ambiguities. So it is either **gu** itself or, when it is not required, nothing at all that terminates a termset.

123:	termset2	=> termset1	Da pa cluva **de fa** = X loved Y **later.**
124:		=> termset2 A termset1	Da pa cluva **de fa, e di pa** = X loved **Y later and W earlier.**
125:		=> KA termset2 KI termset1 Local.	Da pa cluva **ke de fa ki di pa** = X loved **both Y later and W earlier.**
126:	termset	=> termset2	Da sanpa **de di vi do** = X is a sign of **Y** to **W at Q.**
127:		=> PAUSE termset2 128, 139, 141, 148, 154.	Da sanpa, **de di vi do** = X is a sign, of **Y** to **W at Q.** (This discretionary pause-comma in a place where <gap> does not occur must be explicitly provided for.)

Predicates are equipped with individual termsets in R128 and 148, and with joint termsets in R139, 141 and 154. All these applications of <termset> take place in the next group, which constructs predicates. Termsets are used for no other purpose in the grammar than to construct predicates.

Group I. Predicates
Rules 128-154

In this group, the predicate expressions that make the claims of sentences are constructed. Basically they consist of a sentence predicate from Group E coupled with a termset, possibly null, from Group H; this coupling is accomplished in R128. In the sequel we'll call such expressions simply "predicates". Predicates may be "marked" or "bare", that is, prefixed by tense or abstraction operators (which includes the left-marker **ga**) or not so prefixed; see R129-31. This distinction is fundamental; it involved building two tracks through the grammar of predicates, one for bare forms, one for marked ones. For example, in a long sequence of rules, R132-42, a distinction between "backpreds" and the two kinds of "front" predicates, "barefronts" and "markfronts", is gradually developed; and in R137 and 140, this distinction is finally used. It turns out that the forms called backpreds are going to be used as the right connectands of both kinds of ekked connections, the "bareekpreds" and "markekpreds". These are the bare and marked versions, respectively, of ekked predicates, that is, of connected predicates in which the connections are made with afterthought or A-form connectives. Barefronts and markfronts, of course, are destined to be the left or leading connectands of these same ekked predicates; and the ekked connections themselves will of course reflect the "bareness" type of their leading elements.

By R143-54, the bare vs. marked distinction is no longer important. Identity predicates ("identpreds") are now developed--BI and kin may not be treated as "just another PREDA" because they enter into little word compounds and must be recognized by the lexer--and finally kekked, or forethoughtfully connected, predicates ("kekpred") are developed. Along the way, in R135-6 and 146-7, the ACI- and AGE-forms of afterthought connections are developed in the same way that they were developed for arguments.

128:	barepred	=> sentpred termset Local.	Da **fumna** = **X is a woman.** (Here we are using the null termset <gu>, which, because it is final, may be expressed by <err>, i.e., omitted altogether.)
129:	markpred	=> M7 PA barepred	Da **pa fumna** = X **was a woman.**
130:		=> PO gap sentence gap	Da **po de fumna** = **X is a case of Y's being a woman.** (Both <gap>s may be null here, the 2nd because it is final, the 1st because there is no following <predunit> for the normally close-binding 'po' to bind to. **Po** will not "stick" to **de**, and so does not need to be separated from it.)
131:		=> ZO gap sentence gap Local.	Da **zo de hatro** = **X is the amount of heat in Y** (by which Y is hot). (<gap>s as in S130.)
132:	backpred1	=> barepred	Da **fumna, e ditca** = **X is a woman** and **a teacher.** (Both these connectands are barepreds.)
133:		=> markpred	Da fumna, e **pa ditca** = X is a woman and **was a teacher.**
134:		=> M8 NO backpred1 Local.	Da fumna, e **no ga blanu marpi** = X is a woman and **not a blue snake.** (Here **ga** extends

			the scope of 'no' over the whole backpred. Ge would also work but is bad usage.)
135:	backpred	=> backpred1	Da fumna, e ditca = X is a woman and a teacher. (Backpreds are always right connectands.)
136:		=> backpred1 M9 ACI backpred Local.	Da fumna, e ditca, aci stude = X is a woman, and a teacher or student. (Aci binds ditca aci stude into the right connectand of e, thus altering the normal left-grouping of afterthought connection.)
137:	bareekpred	=>barefront M10 A backpred Local.	Da fumna, e ditca, aci stude = X is a woman, and a teacher or student. (This is a bare "ekpred" because its leading element is bare.)
138:	barefront	=> barepred	Da ditca lo dotca = X is a teacher of German.
139:		=> bareekpred termset Local.	Da ditca, e stude gu lo dotca = X is a teacher of, and a student of, German. (Again gu prepares for the joint termset.)
140:	markekpred	=> markfront M10 A backpred Local.	Da pa ditca, e cluva = X was a teacher and a lover. (This is a marked "ekpred" because its leading element is marked.)
141:	markfront	=> markpred	Da pa ditca lo frasa = X was a teacher of French.
142:		=> markekpred termset Local.	Da pa ditca, e cluva gu lo frasa = X was a teacher of, and a lover of, French.
143:	predicate2	=> barefront	Da ditca = X is a teacher.
144:		=> markfront	Da pa ditca = X was a teacher.
145:		=> M8 NO predicate2 Local.	Da no ga blanu tcaro = X is not a blue car. (This is long-scope negation. X may not be a car of any kind.)
146:	predicate1	=> predicate2	Ditca = Be a teacher!
147:		=> predicate2 M11 AGE predicate1 Local.	Gudbi, ege ckano, a briga =

Be good, and be kind or brave! (If e replaced ege, this would read 'Be good and kind, or be brave!')

148: identpred => BI termset
Da bi le mrenu jio pa godzi la Frans = X is the man who went to France.

149: => NO identpred
Local.
Da no bi le mrenu jio pa godzi la Frans = X is not the man who went to France.

150: kekpred => M3 KA predicate KI predicate
Ke ckano ki briga = Be both kind and brave!

151: => NO kekpred
Local.
No ke ckano ki bunbo = Don't be both kind and a fool!

152: predicate => predicate1
Da mrenu = X is a man.

153: => identpred
Da bi le mrenu = X is the man.

154: => kekpred termset
155, 158, 162.
Da ke briga ki bunbo gu raba = X is both brave (about everything) and a fool about everything.

Predicates, once developed, are used in constructing sentences. All three of these references are to the next group, in which sentences are made. In R155 <predicate> is used to make the Goa-sentence; in R158 it is part of the declarative sentence in normal word order; and in R162 it is the whole of the Loglan imperative.

Group J. Sentences
Rules 155-164

This small group of rules catalogues the kinds of Loglan sentences, and gives their formulas. Starting with Goa-sentences (R155), it moves on to the various forms of statements (R156-9), and finally to kekked sentences (R160-1). Finally in R162-4, imperatives, statements and kekked sentences--a variety of what are called "compound sentences" in English grammar--are bundled together into the <sentence> grameme.

155: goasent => GOA predicate GOA terms
Local.
Goa seidjo le so fumna goa to mrenu = Among the six women are two men. (This is the V-O-S word-order which is standard in some languages. It is a heavily marked form in Loglan.)

156: statement	=> goasent	**Goa seidjo Bai goa Cai = Included in B is C.**
157:	=> terms goasent	**Vi levi kruma goa seidjo Bai goa Cai = In this room, included in B is C.** (A variety of the Goa-sentence with fronted modifiers.)
158:	=> terms predicate	**Da fumna = X is a woman.** (This is the Loglan-normal word-order, S-V-O. Here usage restricts the expression of <terms> to a single argument, the subject, followed by none or more modifiers.)
159:	=> NO statement Local.	**No da fumna = It is not the case that X is a woman.**
160: keksent	=> KA sentence KI sentence	**Ko da bi lio ne ki sumduo de = If and only if X is 1, add Y.**
161:	=> NO keksent Local.	**No da ko bi lio ne ki sumduo de = It is not the case that if and only if X is 1, (you should) add Y.**
162: sentence	=> predicate	**Fumna = Be a woman!**
163:	=> statement	**Da fumna = X is a woman.**
164:	=> keksent 174-5; 130-1, 90, 87.	**Kanoi ti bi la Fomen, ki fumna = If this is April, be a woman!**

The two forward references to <sentence> in R174-5 carry sentences into the next group, where they are only one among the many kinds of utterances. But the back references of <sentence> are more interesting. R130-1 show how sentences may be nested inside predicates by means of the abstraction operators **po pu** and **zo**; R90 shows how a sentence may be nested in an argument modifier by using the operators **jia** and **jio** (which produce the Loglan equivalents of 'John, who was the man who came to dinner' and similar constructions); and R87, which permits sentences to become the basis of event descriptions ('Before John came to dinner' and similar clauses). In these three ways, sentences may be embedded in sentences, the embedding sentences embedded in further sentence, and so on, ad infinitum.

Before moving on to utterances it may be worth a moment to consider how it came about that the two instances of <terms> in R157-8 are allowed by usage to have such different expressions. In R157 it is not good usage--in fact, it would not even be interpretable--to include any arguments at all in the optional string of fronted modifiers which usage allows <terms> to be as a possible embellishment of the Goa-sentence. In contrast, usage expects to find something very different as an expression of <terms> in R158. Here, <terms> is the subject of the Loglan-normal sentence; and so it must consist of exactly one argument accompanied by any number of, including zero, following modifiers. Thus, R157 and R158 generate two massive supersets of possibilities only some of which are good usage. Indeed a very different set in each domain is allowed by usage.

The reader may be interested to learn the reason why these supersets appear at just this point in the grammar. They were fashioned as an alternative to writing an extremely long and awkward, and probably always imperfect, lookahead-extending procedure into the preparser...or rather, to abandon one that had already been written. It was first thought that such a procedure could detect without parsing whether a set of early terms was fronted, or fragmentary, or the subject of a declarative sentence plus some early modifiers, or so on; and that the M-Lexemes inserted by that procedure would then permit a grammar of sentences to be written which was devoid of these supersets. But on further study of this matter the Academy came to the conclusion that the hope of writing such a lookahead-extending algorithm was largely illusory; that exceptional cases could probably always be found that could not be handled by it; and that therefore it was preferable from the standpoint of computer science to adopt the superset solution to the "fragment handling" problem rather than attempt to solve it by elaborating the preparser algorithm in new and dubious directions.

Besides, there are good theoretical reasons to believe that just such supersets are part of every natural grammar, and that it is the job of an entirely different kind of rule, the usage rule, to declare which sub-domains in the domains of the parsible are in this higher sense allowable. If so, then what we are doing with Loglan at this point is an ancient linguistical move which may be performed in all languages in approximately the same way. What is most important about this solution is that it solves the problem of parsing fragmentary utterances within the grammar, thus preserving the syntactic unambiguity of the language without dependence on the presumed completeness of a preparser algorithm.

Group K. Utterances
Rules 165-194

Group K is the last group of rules in Loglan grammar. In it are shown all the varieties of utterances, including sentence fragments like answers and monosyllabic responses (which occasion the supersets of Group J) as well as sentences themselves. In addition, the two major elaborations of sentences are provided in this group, namely (1) leading modifiers like **Na la Fomen, gi** in **Na la Fomen, gi mia pa godzi la Europas** and/or fronted arguments like **Dio da gi** in **Dio da gi de pa takna** = 'To X, Y spoke', and (2) the so-called "prenex quantifiers" with which logicians embellish certain kinds of claims: **Ba rabe goi** in **Ba rabe goi ba kunci be** = 'There is an x such that for every y, x is related to y' ('Someone is related to everyone'). Both kinds of elaborations involve the <terms> grameme from Group H, and are carried out in R165-6 and R175.

165: headterms	=>terms GI	**Dio da piu de gi** di pa takna = **To X about Y,** W spoke. (**Dio** and **piu** are 2 of the 11 optional case tags; **dio** is derived from **dirco** and means that X is a "beneficiary or destination" of some kind; **piu** is from **pisku** and means that Y is a kind of "part, passive or topic".)
166:	=>headterms terms GI	**Ba rabe goi dio be gi** ba takna = **There is an x such that for every y, to y,** x talks. (**Goi** is a 2nd allolex of GI and fronts the "prenex quantifiers" of logicians'

talk. This is an unusual word-order, but may be used.)

Usage requires that instances of <terms> used with the prenex fronting operator **goi** be limited to strings of arguments, that is, that there be no modifiers mixed in. Moreover, such arguments must obviously be matchable, except for their quantifiers, with arguments occurring in the body of the sentence to which such headterms are attached; see R175. Thus, the **ba** in **raba** is matchable with the second instance of **ba** in S166. In other words, when **goi** is the fronting operator, the grammatical possibilities opened up by Rules 165-6 form a superset of the forms accepted by usage. Usage and grammar coincide, however, when **gi** does the fronting. <Terms> used with the fronting operator **gi** in these rules may be strings of arguments and/or modifiers in any mixture.

167:	uttA	=> A	**Enoi = And-not** (the first but not the second). (An answer to a HA-question. The monosyllabic allolexes of A, namely, **a e o u,** are also used in spelling. See R191 for **i.**)
168:		=> NO	**No = No.** (Answer to an **Ei**-question.)
169:		=> IE	**Ie = Which?**
170:		=> mex	**Ne. To. Te. Fo = One. Two. Three. Four.** (Used either to answer **Ho**-questions--'How many?'--or in counting.)
171:		=> terms	**La Djan, na la Formen = John, in April.** (A pair of answers to two **Hu**-questions: 'Who? When?' It is the utterance fragments produced by this rule that creates the lookahead problem solved by the "superset solution" in R157-8. Calling subjects, fronted arguments, and answers <terms> relieves the LR1 parser of the necessity of making finer-grained decisions until it is deeper into the parse.)
172:		=> links	**Jue lemi sorme = Through my sister.** (An answer to **Jue hu**-questions.)
173:		=> linkargs	**Je le fotpa botci jue lemi sorme = By the fat boy through my sister.** (An answer to **Je hu**-questions.)
174:		=> sentence	**La Pit, pa takna = Pete talked.**
175:	Local.	=> headterms sentence	**Dio la Djan, na lepo la Ruprt, pa kamla gi la Pit, pa takna = To John, when Rupert came, Pete talked.**

176:	headmod	=> UI	**Ue** nenkaa = **Well,** come in!
177:		=> HOI (name/descriptn,)	**Hoi Redro Nu Herfa,** nenkaa = **O Red-Haired One,** come in! (The name or description is parsed re-entrantly after the primary parse is concluded.)
178:		=> KIE (utterance KIU)	**Kie pazu kiu** ti crina = **(At last)** it rained here.
179:		=> name Local.	**Djan,** nenkaa = **John,** come in!

The four allograms of <headmod> are also the "free modifiers" of Loglan. These are the elements that may occur anywhere in a Loglan utterance, and are taken to "modify"--if that interpretation is at all meaningful--the immediately preceding word, or, if initial, the utterance as a whole. Precisely because these objects may occur anywhere, their occurrence at any given location gives the parser no information. Therefore, their incidence constitutes a kind of "grammatical noise". So except for these headmods--which are the first elements in any utterance-initial strings of them--they are removed by the preparser before the utterance is "handed to" the parser. Thus, the parser never sees attitudinal words like **ia** ('certainly') or discursives like **pou** ('however') unless one occurs at the head of some utterance. Moreover, it never sees names used vocatively, such as **Djan** without a preceding **la,** unless they too are initial in some utterance. This is the preferred position for a free modifier, by the way, because then it is taken as modifying the utterance as a whole...which is usually what is intended. Finally, except for parsing them reentrantly, i.e., when the parse of the main utterance is concluded, the parser never sees descriptive expressions used vocatively with **hoi,** or parenthetic expressions marked by **kie** and **kiu,** unless these, too, are initial in some utterance. For reentrant parsing, see the comment after the last rule in the grammar, R194.

180:	uttB	=> uttA	**Nenkaa = Come in!**
181:		=> headmod	**Djan = John!**
182:		=> headmod uttA Local.	**Pidr, zvosto = Peter, stay out!**
183:	neghead	=> NO gap	**No,** na la Fomen, gi da pa kamla = **It is not the case that** in April, X came.
184:		=> headmod NO gap Local.	**Ui no,** na la Fomen, gi da pa kamla = **Happily it is not the case that** in April, X came.
185:	uttC	=> uttB	**Na la Fomen, gi da pa kamla = In April, X came.**
186:		=> neghead uttC Local.	**Ui no, na la Fomen, gi da pa kamla = Happily it is not the case that in April, X came.**
187:	uttD	=> uttC	**De prano = Y runs.**

188:		=> uttC ICI uttD Local.	Da sucmi, ice de prano, icanoici di fleti = X swims, and Y runs if W flies. (Again the -ci suffix binds the 2 final sentences into the right connectand of ice.)
189:	uttE	=> uttD	Di fleti = W flies.
190:		=> uttE I uttD Local.	Da sucmi, ice de prano, icanoi di fleti = X swims and Y runs, if W flies. (This is the normal left-grouping of afterthought connection without the special grouping effects of -ci or -ge.)
191:	utterance	=> I	Fei. Lei. E. Tei. I = Ef. El. Ee. Tee. Eye. (This rule is used only to parse and generate the letter i when used in spelling.)
192:		=> uttE	De prano = Y runs.
193:		=> I uttE	Icanoi di fleti = If W flies.
194:		=> uttE IGE utterance	Da sucmi, ige de prano, icanoi di fleti = X swims; and Y runs if W flies. (Again the -ge suffix groups all subsequent connected utterances into a single right connectand of the connective so-altered.)

The <utterance> grameme is also used covertly--i.e., carried invisibly through the parse--by KIE in R180 and by LI in R84. These are parenthetic expressions and bits of quoted Loglan, respectively, and like the names and descriptive vocatives of R179, such embedded utterances are parsed "reentrantly". This means that after the parse of an embedding utterance is complete the parser will return to parse the embedded expression, and will continue to do so until all embedded expressions have been parsed.