The Incomplete Lojban Language

Chrestomathy included

John Woldemar Cowan

An unofficial publication, community edition (not by the LLG)

Version geklojban-1.2.16, Generated 2024-10-12

Table of Contents

Chapter 1
Formal grammars

1.1 YACC Grammar of Lojban

The following two listings constitute the formal grammar of Lojban. The first version is written in the YACC language, which is used to describe parsers, and has been used to create a parser for Lojban texts. This parser is available from the Logical Language Group. The second listing is in Extended Backus-Naur Form (EBNF) and represents the same grammar in a more human-readable form. (In case of discrepancies, the YACC version is official.) There is a cross-reference listing for each format that shows, for each selma'o and rule, which rules refer to it.

The Lojban machine parsing algorithm is a multi-step process. The YACC machine grammar presented here is an amalgam of those steps, concatenated so as to allow YACC to verify the syntactic ambiguity of the grammar. YACC is used to generate a parser for a portion of the grammar, which is LALR1 (the type of grammar that YACC is designed to identify and process successfully), but most of the rest of the grammar must be parsed using some language-coded processing.

Step 1 – Lexing

From phonemes, stress, and pause, it is possible to resolve Lojban unambiguously into a stream of words. Any machine processing of speech will have to have some way to deal with “non-Lojban” failures of fluent speech, of course. The resolved words can be expressed as a text file using Lojban's phonetic spelling rules.

The following steps assume that there is the possibility of non-Lojban text within the Lojban text (delimited appropriately). Such non-Lojban text may not be reducible from speech phonetically. However, step 2 allows the filtering of a phonetically transcribed text stream, to recognize such portions of non-Lojban text where properly delimited, without interference with the parsing algorithm.

Step 2 – Filtering

From start to end, performing the following filtering and lexing tasks using the given order of precedence in case of conflict:

i. If the Lojban word *zoi* (selma'o ZOI) is identified, take the following Lojban word (which should be end delimited with a pause for separation from the following non-Lojban text) as an opening delimiter. Treat all text following that delimiter, until that delimiter recurs *after a pause*, as

grammatically a single token (labelled "YACC rule #699 (p. 8)" in this grammar). There is no need for processing within this text except as necessary to find the closing delimiter.

ii. If the Lojban word *zo* (selma'o ZO) is identified, treat the following Lojban word as a token labelled "YACC rule #698 (p. 8)", instead of lexing it by its normal grammatical function.

iii. If the Lojban word *lo'u* (selma'o LOhU) is identified, search for the closing delimiter *le'u* (selma'o LEhU), ignoring any such closing delimiters absorbed by the previous two steps. The text between the delimiters should be treated as the single token "YACC rule #697 (p. 8)".

iv. Categorize all remaining words into their Lojban selma'o category, including the various delimiters mentioned in the previous steps. In all steps after step 2, only the selma'o token type is significant for each word.

v. If the word *si* (selma'o SI) is identified, erase it and the previous word (or token, if the previous text has been condensed into a single token by one of the above rules).

vi. If the word *sa* (selma'o SA) is identified, erase it and all preceding text as far back as necessary to make what follows attach to what precedes. (This rule is hard to formalize and may receive further definition later.)

vii. If the word *su* (selma'o SU) is identified, erase it and all preceding text back to and including the first preceding token word which is in one of the selma'o: NIhO, LU, TUhE, and TO. However, if speaker identification is available, a SU shall only erase to the beginning of a speaker's discourse, unless it occurs at the beginning of a speaker's discourse. (Thus, if the speaker has said something, two adjacent uses of *su* are required to erase the entire conversation.

Step 3 – Termination

If the text contains a FAhO, treat that as the end-of-text and ignore everything that follows it.

Step 4 – Absorption of Grammar-Free Tokens

In a new pass, perform the following absorptions (absorption means that the token is removed from the grammar for processing in following steps, and optionally reinserted, grouped with the absorbing token after parsing is completed).

i. Token sequences of the form any - (ZEI - any) ..., where there may be any number of ZEIs, are merged into a single token of selma'o BRIVLA.

ii. Absorb all selma'o BAhE tokens into the following token. If they occur at the end of text, leave them alone (they are errors).

iii. Absorb all selma'o BU tokens into the previous token. Relabel the previous token as selma'o BY.

iv. If selma'o NAI occurs immediately following any of tokens UI or CAI, absorb the NAI into the previous token.

v. Absorb all members of selma'o DAhO, FUhO, FUhE, UI, Y, and CAI into the previous token. All of these null grammar tokens are permitted following any word of the grammar, without interfering with that word's grammatical function, or causing any effect on the grammatical interpretation of any other token in the text. Indicators at the beginning of text are explicitly handled by the grammar.

Step 5 – Insertion of Lexer Lexemes

Lojban is not in itself LALR1. There are words whose grammatical function is determined by following tokens. As a result, parsing of the YACC grammar must take place in two steps. In the first step, certain strings of tokens with defined grammars are identified, and either

i. are replaced by a single specified "lexer token" for step 6, or

ii. the lexer token is inserted in front of the token string to identify it uniquely.

The YACC grammar included herein is written to make YACC generation of a step 6 parser easy regardless of whether a. or b. is used. The strings of tokens to be labelled with lexer tokens are found in rule terminals labelled with numbers between 900 and 1099. These rules are defined with the lexer tokens inserted, with the result that it can be verified that the language is LALR1 under option b. after steps 1 through 4 have been performed. Alternatively, if option a. is to be used, these rules are commented out, and the rule terminals labelled from 800 to 900 refer to the lexer tokens *without* the strings of defining tokens. Two sets of lexer tokens are defined in the token set so as to be compatible with either option.

In this step, the strings must be labelled with the appropriate lexer tokens. Order of inserting lexer tokens *IS* significant, since some shorter strings that would be marked with a lexer token may be found inside longer strings. If the tokens are inserted before or in place of the shorter strings, the longer strings cannot be identified.

If option a. is chosen, the following order of insertion works correctly (it is not the only possible order): A, C, D, B, U, E, H, I, J, K, M, N, G, O, V, W, F, P, R, T, S, Y, L, Q. This ensures that the longest rules will be processed first; a PA+MAI will not be seen as a PA with a dangling MAI at the end, for example.

Step 6 – YACC Parsing

YACC should now be able to parse the Lojban text in accordance with the rule terminals labelled from 1 to 899 under option 5a, or 1 to 1099 under option 5b. Comment out the rules beyond 900 if option 5a is used, and comment out the 700-series of lexer-tokens, while restoring the series of lexer tokens numbered from 900 up.

```
%token A_501            eks; basic afterthought logical connectives
%token BAI_502          modal operators
%token BAhE_503         next word intensifier
%token BE_504           sumti link to attach sumti to a selbri
%token BEI_505          multiple sumti separator between BE, BEI
%token BEhO_506         terminates BE/BEI specified descriptors
%token BIhI_507         interval component of JOI
%token BO_508           joins two units with shortest scope
%token BRIVLA_509       any brivla
%token BU_511           turns any word into a BY lerfu word
%token BY_513           individual lerfu words
%token CAhA_514         specifies actuality/potentiality of tense
%token CAI_515          afterthought intensity marker
%token CEI_516          pro-bridi assignment operator
%token CEhE_517         afterthought term list connective
%token CMENE_518        Lojbanized names; require consonant end, as well as
%token CO_519           tanru inversion
%token COI_520          vocative marker permitted inside cmevla; must always
%token CU_521           separator between head sumti and selbri
%token CUhE_522         tense/modal question
%token DAhO_524         cancel anaphora/cataphora assignments
%token DOI_525          vocative marker
%token DOhU_526         terminator for DOI-marked vocatives
%token FA_527           modifier head generic case tag
%token FAhA_528         superdirections in space
%token FAhO_529         normally elided “done pause” to indicate end of utte
%token FEhE_530         space interval mod flag
%token FEhU_531         ends bridi to modal conversion
%token FIhO_532         marks bridi to modal conversion
%token FOI_533          end compound lerfu
%token FUhE_535         open long scope for indicator
%token FUhO_536         close long scope for indicator
%token GA_537           geks; forethought logical connectives
%token GEhU_538         marker ending GOI relative clauses
%token GI_539           forethought medial marker
%token GIhA_541         logical connectives for bridi-tails
%token GOI_542          attaches a sumti modifier to a sumti
%token GOhA_543         pro-bridi
%token GUhA_544         GEK for tanru units, corresponds to JEKs
%token I_545            sentence link
%token JA_546           jeks; logical connectives within tanru
%token JAI_547          modal conversion flag
%token JOI_548          non-logical connectives
%token KEhE_550         right terminator for KE groups
%token KE_551           left long scope marker
%token KEI_552          right terminator, NU abstractions
%token KI_554           multiple utterance scope for tenses
%token KOhA_555         sumti anaphora
%token KU_556           right terminator for descriptions, etc.
%token KUhO_557         right terminator, NOI relative clauses
%token LA_558           name descriptors
%token LAU_559          lerfu prefixes
%token LAhE_561         sumti qualifiers
%token LE_562           sumti descriptors
```

```
%token LEhU_565          possibly ungrammatical text right quote
%token LI_566            convert number to sumti
%token LIhU_567          grammatical text right quote
%token LOhO_568          elidable terminator for LI
%token LOhU_569          possibly ungrammatical text left quote
%token LU_571            grammatical text left quote
%token LUhU_573          LAhE close delimiter
%token ME_574            converts a sumti into a tanru_unit
%token MEhU_575          terminator for ME
%token MOhI_577          motion tense marker
%token NA_578            bridi negation
%token NAI_581           attached to words to negate them
%token NAhE_583          scalar negation
%token NIhO_584          new paragraph; change of subject
%token NOI_585           attaches a subordinate clause to a sumti
%token NU_586            abstraction
%token NUhI_587          marks the start of a termset
%token NUhU_588          marks the middle and end of a termset
%token PEhE_591          afterthought termset connective prefix
%token PU_592            directions in time
%token RAhO_593          flag for modified interpretation of GOhI
%token ROI_594           converts number to extensional tense
%token SA_595            metalinguistic eraser to the beginning of the curren
%token SE_596            conversions
%token SEI_597           metalinguistic bridi insert marker
%token SEhU_598          metalinguistic bridi end marker
%token SI_601            metalinguistic single word eraser
%token SOI_602           reciprocal sumti marker
%token SU_603            metalinguistic eraser of the entire text
%token TAhE_604          tense interval properties
%token TEI_605           start compound lerfu
%token TO_606            left discursive parenthesis
%token TOI_607           right discursive parenthesis
%token TUhE_610          multiple utterance scope mark
%token TUhU_611          multiple utterance end scope mark
%token UI_612            attitudinals, observationals, discursives
%token VA_613            distance in space-time
%token VAU_614           end simple bridi or bridi-tail
%token VEhA_615          space-time interval size
%token VIhA_616          space-time dimensionality marker
%token VUhO_617          glue between logically connected sumti  and relative
%token XI_618            subscripting operator
%token Y_619             hesitation
%token ZAhO_621          event properties – prospective, etc.
%token  ZEhA_622          time interval size tense
%token ZEI_623           lujvo glue
%token ZI_624            time distance tense
%token ZIhE_625          conjoins relative clauses
%token ZO_626            single word metalinguistic quote marker
%token ZOI_627           delimited quote marker
%token ZOhU_628          prenex terminator (not elidable)
%token BIhE_650          prefix for high-priority MEX operator
%token BOI_651           number or lerfu-string terminator
%token FUhA_655          reverse Polish flag
```

```
%token GAhO_656          open/closed interval markers for BIhI
%token JOhI_657          flags an array operand
%token KUhE_658          MEX forethought delimiter
%token MAI_661           change numbers to utterance ordinals
%token MAhO_662          change MEX expressions to MEX operators
%token MOI_663           change number to selbri
%token MOhE_664          change sumti to operand, inverse of LI
%token NAhU_665          change a selbri into an operator
%token NIhE_666          change selbri to operand; inverse of MOI
%token NUhA_667          change operator to selbri; inverse of MOhE
%token PA_672            numbers and numeric punctuation
%token PEhO_673          forethought (Polish) flag
%token TEhU_675          closing gap for MEX constructs
%token VEI_677           left MEX bracket
%token VEhO_678          right MEX bracket
%token VUhU_679          MEX operator
%token any_words_697     a string of lexable Lojban words
%token any_word_698      any single lexable Lojban words
%token anything_699      a possibly unlexable phoneme string
The following tokens are the actual lexer tokens.  The _900 series tokens are
%token lexer_ A_701       flags a MAI utterance ordinal
%token lexer_ B_702       flags an EK unless EK_BO, EK_KE
%token lexer_ C_703       flags an EK_BO
%token lexer_ D_704       flags an EK_KE
%token lexer_ E_705       flags a JEK
%token lexer_ F_706       flags a JOIK
%token lexer_ G_707       flags a GEK
%token lexer_ H_708       flags a GUhEK
%token lexer_ I_709       flags a NAhE_BO
%token lexer_ J_710       flags a NA_KU
%token lexer_ K_711       flags an I_BO (option. JOIK/JEK lexer tags)
%token lexer_ L_712       flags a PA, unless MAI (then lexer A)
%token lexer_ M_713       flags a GIhEK_BO
%token lexer_ N_714       flags a GIhEK_KE
%token lexer_ O_715       flags a modal operator BAI or compound
%token lexer_ P_716       flags a GIK
%token lexer_ Q_717       flags a lerfu_string unless MAI (then lexer_A)
%token lexer_ R_718       flags a GIhEK, not BO or KE
%token lexer_ S_719       flags simple I
%token lexer_ T_720       flags I_JEK
%token lexer_ U_721       flags a JEK_BO
%token lexer_ V_722       flags a JOIK_BO
%token lexer_ W_723       flags a JOIK_KE
%token lexer_ X_724    null
%token lexer_ Y_725       flags a PA_MOI
%token lexer_A_905     :  lexer_A_701  utt_ordinal_root_906
%token lexer_B_910     :  lexer_B_702  EK_root_911
%token lexer_C_915     :  lexer_C_703  EK_root_911  BO_508
%token lexer_D_916     :  lexer_D_704  EK_root_911  KE_551
%token lexer_E_925     :  lexer_E_705  JEK_root_926
%token lexer_F_930     :  lexer_F_706  JOIK_root_931
%token lexer_G_935     :  lexer_G_707  GA_537
%token lexer_H_940     :  lexer_H_708  GUhA_544
%token lexer_I_945     :  lexer_I_709  NAhE_583  BO_508
```

```
%token lexer_J_950     :  lexer_J_710  NA_578  KU_556
%token lexer_K_955     :  lexer_K_711  I_432  BO_508
%token lexer_L_960     :  lexer_L_712  number_root_961
%token lexer_M_965     :  lexer_M_713  GIhEK_root_991  BO_508
%token lexer_N_966     :  lexer_N_714  GIhEK_root_991  KE_551
%token lexer_O_970     :  lexer_O_715  simple_tense_modal_972
%token lexer_P_980     :  lexer_P_716  GIK_root_981
%token lexer_Q_985     :  lexer_Q_717  lerfu_string_root_986
%token lexer_R_990     :  lexer_R_718  GIhEK_root_991
%token lexer_S_995     :  lexer_S_719  I_545
%token lexer_T_1000    :  lexer_T_720  I_545  simple_JOIK_JEK_957
%token lexer_U_1005    :  lexer_U_721  JEK_root_926  BO_508
%token lexer_V_1010    :  lexer_V_722  JOIK_root_931  BO_508
%token lexer_W_1015    :  lexer_W_723  JOIK_root_931  KE_551
%token lexer_X_1020    null
%token lexer_Y_1025    :  lexer_Y_725  number_root_961  MOI_663
%startYACC rule #0 (p. 9)
%% text_0                   :
YACC rule #1 (p. 9)
                         |
YACC rule #411 (p. 26)
YACC rule #1 (p. 9)
                         |
YACC rule #32 (p. 12)
YACC rule #1 (p. 9)
                         |
YACC rule #404 (p. 26)
YACC rule #1 (p. 9)
                         |
YACC rule #411 (p. 26)
YACC rule #32 (p. 12)
YACC rule #1 (p. 9)
                         |
YACC rule #581 (p. 7)
YACC rule #0 (p. 9)
                         ;
 text_A_1                 :
YACC rule #422 (p. 27)
YACC rule #2 (p. 9)
                          incomplete JOIK_JEK  without preceding I
                          compare note on
YACC rule #10 (p. 10)
                         |
YACC rule #2 (p. 9)
                         ;
 text_B_2                 :
YACC rule #819 (p. 39)
YACC rule #2 (p. 9)
                         |
YACC rule #820 (p. 39)
YACC rule #2 (p. 9)
                         |
YACC rule #811 (p. 38)
YACC rule #2 (p. 9)
```

```
                        |
YACC rule #410 (p. 26)
YACC rule #3 (p. 10)
                        |
YACC rule #3 (p. 10)
                        ;
 text_C_3                :
YACC rule #4 (p. 10)
    Only indicators which follow certain selma'o:
    cmevla,
YACC rule #607 (p. 7),
YACC rule #571 (p. 7), and the lexer_K and lexer_S I_roots and compounds,
    and at the start of text(_0), will survive the lexer; all other valid one
    will be absorbed.  The only strings for which indicators generate a
    potential ambiguity are those which contain NAI.  An indicator cannot be
    inserted in between a token and its negating NAI, else you can't tell
    whether it is the indicator or the original token being negated.
                        |   empty
   An empty text is legal; formerly this was handled by the explicit
   appearance of
YACC rule #529 (p. 6), but this is now absorbed by the preparser.
                        ;

 paragraphs_4           :
YACC rule #10 (p. 10)
                        |
YACC rule #10 (p. 10)
YACC rule #410 (p. 26)
YACC rule #4 (p. 10)
                        ;
 paragraph_10           :
YACC rule #11 (p. 10)
                        |
YACC rule #20 (p. 11)
                        |
YACC rule #10 (p. 10)
YACC rule #819 (p. 39)
YACC rule #11 (p. 10)
                        |
YACC rule #10 (p. 10)
YACC rule #819 (p. 39)
YACC rule #20 (p. 11)
                        |
YACC rule #10 (p. 10)
YACC rule #819 (p. 39)
                         this last fixes an erroneous start to a sentence,
                           and permits incomplete JOIK_JEK after I, as well
                           in answer to questions on those connectives
                        ;
 statement_11           :
YACC rule #12 (p. 11)
                        |
YACC rule #30 (p. 12)
YACC rule #11 (p. 10)
```

```
                        ;
 statement_A_12          :
YACC rule #13 (p. 11)
                        |
YACC rule #12 (p. 11)
YACC rule #820 (p. 39)
YACC rule #13 (p. 11)
                        |
YACC rule #12 (p. 11)
YACC rule #820 (p. 39)
                        ;
 statement_B_13          :
YACC rule #14 (p. 11)
                        |
YACC rule #14 (p. 11)
YACC rule #811 (p. 38)
YACC rule #13 (p. 11)
                        |
YACC rule #14 (p. 11)
YACC rule #811 (p. 38)
                        ;
 statement_C_14          :
YACC rule #40 (p. 13)
                        |
YACC rule #447 (p. 31)
YACC rule #2 (p. 9)
YACC rule #454 (p. 32)
                        |
YACC rule #491 (p. 36)
YACC rule #447 (p. 31)
YACC rule #2 (p. 9)
YACC rule #454 (p. 32)
                        ;

 fragment_20             :
YACC rule #802 (p. 38)
                        |
YACC rule #445 (p. 31)
                        |
YACC rule #818 (p. 39)
                        |
YACC rule #300 (p. 21)
                        |
YACC rule #80 (p. 15)
YACC rule #456 (p. 32)         answer to ma

YACC rule #490 (p. 36) requires both
YACC rule #450 (p. 31) and
YACC rule #456 (p. 32)
                           but needs no extra rule to accomplish this
                        |
YACC rule #121 (p. 19)
                        |
YACC rule #161 (p. 21)
```

```
                        |
YACC rule #160 (p. 21)
                        |
YACC rule #30 (p. 12)
                        ;

 prenex_30               :
YACC rule #80 (p. 15)
YACC rule #492 (p. 37)
                        ;
 free_modifier_32        :
YACC rule #33 (p. 12)
                        |
YACC rule #33 (p. 12)
YACC rule #32 (p. 12)
                        ;
 free_modifier_A_33      :
YACC rule #35 (p. 12)
                        |
YACC rule #36 (p. 13)
                        |
YACC rule #34 (p. 12)
                        |
YACC rule #486 (p. 36)
                        |
YACC rule #801 (p. 38)
                        ;
 discursive_bridi_34     :
YACC rule #440 (p. 30)
YACC rule #130 (p. 19)
YACC rule #459 (p. 32)
                        |
YACC rule #498 (p. 37)
YACC rule #90 (p. 16)
YACC rule #459 (p. 32)
                        |
YACC rule #498 (p. 37)
YACC rule #90 (p. 16)
YACC rule #90 (p. 16)
YACC rule #459 (p. 32)
                        |
YACC rule #440 (p. 30)
YACC rule #80 (p. 15)
YACC rule #451 (p. 32)
YACC rule #130 (p. 19)
YACC rule #459 (p. 32)
                        |
YACC rule #440 (p. 30)
YACC rule #80 (p. 15)
YACC rule #130 (p. 19)
YACC rule #459 (p. 32)
                        ;
 vocative_35             :
YACC rule #415 (p. 27)
```

```
YACC rule #130 (p. 19)
YACC rule #457 (p. 32)
                          |
YACC rule #415 (p. 27)
YACC rule #130 (p. 19)
YACC rule #121 (p. 19)
YACC rule #457 (p. 32)
                          |
YACC rule #415 (p. 27)
YACC rule #121 (p. 19)
YACC rule #130 (p. 19)
YACC rule #457 (p. 32)
                          |
YACC rule #415 (p. 27)
YACC rule #121 (p. 19)
YACC rule #130 (p. 19)
YACC rule #121 (p. 19)
YACC rule #457 (p. 32)
                          |
YACC rule #415 (p. 27)
YACC rule #404 (p. 26)
YACC rule #457 (p. 32)
                          |
YACC rule #415 (p. 27)
YACC rule #404 (p. 26)
YACC rule #121 (p. 19)
YACC rule #457 (p. 32)
                          |
YACC rule #415 (p. 27)
YACC rule #121 (p. 19)
YACC rule #404 (p. 26)
YACC rule #457 (p. 32)
                          |
YACC rule #415 (p. 27)
YACC rule #121 (p. 19)
YACC rule #404 (p. 26)
YACC rule #121 (p. 19)
YACC rule #457 (p. 32)
                          |
YACC rule #415 (p. 27)
YACC rule #90 (p. 16)
YACC rule #457 (p. 32)
                          |
YACC rule #415 (p. 27)
YACC rule #457 (p. 32)
                          ;
 parenthetical_36          :
YACC rule #606 (p. 7)
YACC rule #0 (p. 9)
YACC rule #468 (p. 34)
                          ;

 sentence_40               :
YACC rule #50 (p. 14)   bare observative or mo answer
```

```
                        |
YACC rule #80 (p. 15)
YACC rule #451 (p. 32)
YACC rule #50 (p. 14)
                        |
YACC rule #80 (p. 15)
YACC rule #50 (p. 14)
                        ;
 subsentence_41          :
YACC rule #40 (p. 13)
                        |
YACC rule #30 (p. 12)
YACC rule #41 (p. 14)
                        ;

 bridi_tail_50           :
YACC rule #51 (p. 14)
                        |
YACC rule #51 (p. 14)
YACC rule #814 (p. 39)
YACC rule #50 (p. 14)
YACC rule #466 (p. 33)
YACC rule #71 (p. 15)
                        ;
 bridi_tail_A_51         :
YACC rule #52 (p. 14)
                        |
YACC rule #51 (p. 14)
YACC rule #818 (p. 39)
YACC rule #52 (p. 14)
YACC rule #71 (p. 15)
                        ;
 bridi_tail_B_52         :
YACC rule #53 (p. 14)
                        |
YACC rule #53 (p. 14)
YACC rule #813 (p. 39)
YACC rule #52 (p. 14)
YACC rule #71 (p. 15)
                        ;
 bridi_tail_C_53         :
YACC rule #54 (p. 14)
                        |
YACC rule #130 (p. 19)
YACC rule #71 (p. 15)
                        ;
 gek_sentence_54         :
YACC rule #807 (p. 38)
YACC rule #41 (p. 14)
YACC rule #816 (p. 39)
YACC rule #41 (p. 14)
YACC rule #71 (p. 15)
                        |
```

```
YACC rule #491 (p. 36)
YACC rule #493 (p. 37)
YACC rule #54 (p. 14)
YACC rule #466 (p. 33)
                        |
YACC rule #445 (p. 31)
YACC rule #54 (p. 14)
                        ;
 tail_terms_71           :
YACC rule #80 (p. 15)
YACC rule #456 (p. 32)
                        |
YACC rule #456 (p. 32)
                        ;

 terms_80                :
YACC rule #81 (p. 15)
                        |
YACC rule #80 (p. 15)
YACC rule #81 (p. 15)
                        ;
 terms_A_81              :
YACC rule #82 (p. 15)
                        |
YACC rule #81 (p. 15)
YACC rule #494 (p. 37)
YACC rule #422 (p. 27)
YACC rule #82 (p. 15)
                        ;
 terms_B_82              :
YACC rule #83 (p. 15)
                        |
YACC rule #82 (p. 15)
YACC rule #495 (p. 37)
YACC rule #83 (p. 15)
                        ;
 term_83                 :
YACC rule #90 (p. 16)
                        |
YACC rule #84 (p. 15)
                        |
YACC rule #85 (p. 15)
                        |
YACC rule #810 (p. 38)
                        ;
 modifier_84             :
YACC rule #490 (p. 36)
YACC rule #450 (p. 31)
                        |
YACC rule #490 (p. 36)
YACC rule #90 (p. 16)
                        ;
 term_set_85             :
YACC rule #496 (p. 37)
```

```
YACC rule #80 (p. 15)
YACC rule #460 (p. 33)
                        |
YACC rule #496 (p. 37)
YACC rule #807 (p. 38)
YACC rule #80 (p. 15)
YACC rule #460 (p. 33)
YACC rule #816 (p. 39)
YACC rule #80 (p. 15)
YACC rule #460 (p. 33)
                        ;

 sumti_90                :
YACC rule #91 (p. 16)
                        |
YACC rule #91 (p. 16)
YACC rule #497 (p. 37)
YACC rule #121 (p. 19)
                        ;
 sumti_A_91              :
YACC rule #92 (p. 16)
                        |
YACC rule #92 (p. 16)
YACC rule #804 (p. 38)
YACC rule #90 (p. 16)
YACC rule #466 (p. 33)
                        |
YACC rule #92 (p. 16)
YACC rule #823 (p. 40)
YACC rule #90 (p. 16)
YACC rule #466 (p. 33)
                        ;
 sumti_B_92              :
YACC rule #93 (p. 16)
                        |
YACC rule #92 (p. 16)
YACC rule #421 (p. 27)
YACC rule #93 (p. 16)
                        ;
 sumti_C_93              :
YACC rule #94 (p. 16)
                        |
YACC rule #94 (p. 16)
YACC rule #803 (p. 38)
YACC rule #93 (p. 16)
                        |
YACC rule #94 (p. 16)
YACC rule #822 (p. 40)
YACC rule #93 (p. 16)
                        ;
 sumti_D_94              :
YACC rule #95 (p. 17)
                        |
YACC rule #807 (p. 38)
```

```
YACC rule #90 (p. 16)
YACC rule #816 (p. 39)
YACC rule #94 (p. 16)
                        ;
 sumti_E_95              :
YACC rule #96 (p. 17)
                        |
YACC rule #96 (p. 17)
YACC rule #121 (p. 19)
                            indefinite sumti


                        |
YACC rule #300 (p. 21)
YACC rule #130 (p. 19)
YACC rule #450 (p. 31)
                        |
YACC rule #300 (p. 21)
YACC rule #130 (p. 19)
YACC rule #450 (p. 31)
YACC rule #121 (p. 19)
                        ;
 sumti_F_96              :
YACC rule #97 (p. 17)
                            outer-quantified sumti


                        |
YACC rule #300 (p. 21)
YACC rule #97 (p. 17)
                        ;
 sumti_G_97              :
YACC rule #483 (p. 35)
YACC rule #90 (p. 16)
YACC rule #463 (p. 33)
                        |
YACC rule #483 (p. 35)
YACC rule #121 (p. 19)
YACC rule #90 (p. 16)
YACC rule #463 (p. 33)
                           sumti grouping, set/mass/individual conversion; al
                        |
YACC rule #400 (p. 25)
                        |
YACC rule #499 (p. 37)
YACC rule #404 (p. 26)
                        |
YACC rule #499 (p. 37)
YACC rule #121 (p. 19)
YACC rule #404 (p. 26)
                        |
YACC rule #489 (p. 36)
YACC rule #310 (p. 22)
```

```
YACC rule #472 (p. 34)
                        |
YACC rule #110 (p. 18)
                        |
YACC rule #432 (p. 29)
                        ;


 description_110         :
YACC rule #499 (p. 37)
YACC rule #111 (p. 18)
YACC rule #450 (p. 31)
                        |
YACC rule #488 (p. 36)
YACC rule #111 (p. 18)
YACC rule #450 (p. 31)
                        ;
 sumti_tail_111          :
YACC rule #112 (p. 18)
                            inner-quantified sumti relative clause




                        |
YACC rule #121 (p. 19)
YACC rule #112 (p. 18)
                            pseudo-possessive
                              (an abbreviated inner restriction);
                              note that sumti cannot be quantified
                        |
YACC rule #97 (p. 17)
YACC rule #112 (p. 18)
                            pseudo-possessive with outer restriction
                        |
YACC rule #97 (p. 17)
YACC rule #121 (p. 19)
YACC rule #112 (p. 18)
                        ;
 sumti_tail_A_112        :
YACC rule #130 (p. 19)
                        |
YACC rule #130 (p. 19)
YACC rule #121 (p. 19)
                            explicit inner quantifier
                        |
YACC rule #300 (p. 21)
YACC rule #130 (p. 19)
                            quantifier both internal to a description,
                              and external to a sumti thereby made specific
                        |
YACC rule #300 (p. 21)
YACC rule #130 (p. 19)
YACC rule #121 (p. 19)
                        |
```

```
YACC rule #300 (p. 21)
YACC rule #90 (p. 16)
                        ;
 relative_clauses_121    :
YACC rule #122 (p. 19)
                        |
YACC rule #121 (p. 19)
YACC rule #487 (p. 36)
YACC rule #122 (p. 19)
                        ;
 relative_clause_122     :
YACC rule #485 (p. 36)
YACC rule #83 (p. 15)
YACC rule #464 (p. 33)
                        |
YACC rule #484 (p. 36)
YACC rule #41 (p. 14)
YACC rule #469 (p. 34)
                        ;

 selbri_130              :
YACC rule #491 (p. 36)
YACC rule #131 (p. 19)
                        |
YACC rule #131 (p. 19)
                        ;
 selbri_A_131            :
YACC rule #132 (p. 19)
                        |
YACC rule #445 (p. 31)
YACC rule #130 (p. 19)
                        ;
 selbri_B_132            :
YACC rule #133 (p. 19)
                        |
YACC rule #133 (p. 19)
YACC rule #443 (p. 31)
YACC rule #132 (p. 19)
                        ;
 selbri_C_133            :
YACC rule #134 (p. 19)
                        |
YACC rule #133 (p. 19)
YACC rule #134 (p. 19)
                        ;
 selbri_D_134            :
YACC rule #135 (p. 20)
                        |
YACC rule #134 (p. 19)
YACC rule #422 (p. 27)
YACC rule #135 (p. 20)
                        |
YACC rule #134 (p. 19)
YACC rule #823 (p. 40)
```

```
YACC rule #133 (p. 19)
YACC rule #466 (p. 33)
                        ;
 selbri_E_135            :
YACC rule #136 (p. 20)
                        |
YACC rule #136 (p. 20)
YACC rule #821 (p. 40)
YACC rule #135 (p. 20)
                        |
YACC rule #136 (p. 20)
YACC rule #822 (p. 40)
YACC rule #135 (p. 20)
                        ;
 selbri_F_136            :
YACC rule #150 (p. 20)
                        |
YACC rule #150 (p. 20)
YACC rule #479 (p. 35)
YACC rule #136 (p. 20)
                        |
YACC rule #137 (p. 20)
                        |
YACC rule #482 (p. 35)
YACC rule #137 (p. 20)
                        ;
 GUhEK_selbri_137        :
YACC rule #808 (p. 38)
YACC rule #130 (p. 19)
YACC rule #816 (p. 39)
YACC rule #136 (p. 20)
                        ;

 tanru_unit_150          :
YACC rule #151 (p. 20)
                        |
YACC rule #150 (p. 20)
YACC rule #444 (p. 31)
YACC rule #151 (p. 20)
                        ;

 tanru_unit_A_151        :
YACC rule #152 (p. 20)
                        |
YACC rule #152 (p. 20)
YACC rule #160 (p. 21)
                        ;
 tanru_unit_B_152        :
YACC rule #407 (p. 26)
                        |
YACC rule #493 (p. 37)
YACC rule #133 (p. 19)
YACC rule #466 (p. 33)
                        |
```

```
YACC rule #480 (p. 35)
YACC rule #152 (p. 20)
                        |
YACC rule #478 (p. 35)
YACC rule #491 (p. 36)
YACC rule #152 (p. 20)
                        |
YACC rule #478 (p. 35)
YACC rule #152 (p. 20)
                        |
YACC rule #477 (p. 35)
YACC rule #90 (p. 16)
YACC rule #465 (p. 33)
                        |
YACC rule #477 (p. 35)
YACC rule #90 (p. 16)
YACC rule #465 (p. 33)
YACC rule #476 (p. 35)
                        |
YACC rule #475 (p. 35)
YACC rule #374 (p. 24)
                        |
YACC rule #482 (p. 35)
YACC rule #152 (p. 20)
                        |
YACC rule #425 (p. 28)
YACC rule #41 (p. 14)
YACC rule #453 (p. 32)
                        ;

 linkargs_160            :
YACC rule #446 (p. 31)
YACC rule #83 (p. 15)
YACC rule #467 (p. 34)
                        |
YACC rule #446 (p. 31)
YACC rule #83 (p. 15)
YACC rule #161 (p. 21)
YACC rule #467 (p. 34)
                        ;
 links_161               :
YACC rule #442 (p. 31)
YACC rule #83 (p. 15)
                        |
YACC rule #442 (p. 31)
YACC rule #83 (p. 15)
YACC rule #161 (p. 21)
                        ;


  Main entry point for MEX; everything but a number must be in parens.
 quantifier_300          :
YACC rule #812 (p. 39)
YACC rule #461 (p. 33)
```

```
                        |
YACC rule #470 (p. 34)
YACC rule #310 (p. 22)
YACC rule #471 (p. 34)
                        ;
```

Entry point for MEX used after LI; no parens needed, but LI now has an elidable terminator. (This allows us to express the difference between “the expression a + b” and “the expression (a + b)”)

This rule supports left-grouping infix expressions and reverse Polish

expressions. To handle infix monadic, use a null operand; to handle

infix with more than two operands (whatever that means) use an extra operator or an array operand.

```
 MEX_310                :
YACC rule #311 (p. 22)
                        |
YACC rule #310 (p. 22)
YACC rule #370 (p. 23)
YACC rule #311 (p. 22)
                        |
YACC rule #441 (p. 31)
YACC rule #330 (p. 23)
                        ;
```

Support for right-grouping (short scope) infix expressions with BIhE.

```
 MEX_A_311              :
YACC rule #312 (p. 22)
                        |
YACC rule #312 (p. 22)
YACC rule #439 (p. 30)
YACC rule #370 (p. 23)
YACC rule #311 (p. 22)
                        ;
```

Support for forethought (Polish) expressions. These begin with a forethought flag, then the operator and then the argument(s).

```
 MEX_B_312              :
YACC rule #381 (p. 24)
                        |
YACC rule #370 (p. 23)
YACC rule #313 (p. 23)
YACC rule #452 (p. 32)
                        |
YACC rule #438 (p. 30)
YACC rule #370 (p. 23)
YACC rule #313 (p. 23)
YACC rule #452 (p. 32)
                        ;
```

```
 MEX_C_313               :
YACC rule #312 (p. 22)
                        |
YACC rule #313 (p. 23)
YACC rule #312 (p. 22)
                        ;


  Reverse Polish expressions always have exactly two operands.
    To handle one operand, use a null operand;

    to handle more than two operands, use a null operator.

 rp_expression_330       :
YACC rule #332 (p. 23)
YACC rule #332 (p. 23)
YACC rule #370 (p. 23)
                        ;

 rp_operand_332          :
YACC rule #381 (p. 24)
                        |
YACC rule #330 (p. 23)
                        ;


  Operators may be joined by logical connectives.
 operator_370            :
YACC rule #371 (p. 23)
                        |
YACC rule #370 (p. 23)
YACC rule #422 (p. 27)
YACC rule #371 (p. 23)
                        |
YACC rule #370 (p. 23)
YACC rule #823 (p. 40)
YACC rule #370 (p. 23)
YACC rule #466 (p. 33)
                        ;
 operator_A_371          :
YACC rule #372 (p. 24)
                        |
YACC rule #808 (p. 38)
YACC rule #371 (p. 23)
YACC rule #816 (p. 39)
YACC rule #372 (p. 24)
                        |
YACC rule #372 (p. 24)
YACC rule #822 (p. 40)
YACC rule #371 (p. 23)
                        |
YACC rule #372 (p. 24)
YACC rule #821 (p. 40)
YACC rule #371 (p. 23)
```

```
                        ;
 operator_B_372          :
YACC rule #374 (p. 24)
                        |
YACC rule #493 (p. 37)
YACC rule #370 (p. 23)
YACC rule #466 (p. 33)
                        ;
 MEX_operator_374        :
YACC rule #679 (p. 8)
                        |
YACC rule #679 (p. 8)
YACC rule #32 (p. 12)
                        |
YACC rule #480 (p. 35)
YACC rule #374 (p. 24)
    changes argument order
                        |
YACC rule #482 (p. 35)
YACC rule #374 (p. 24)
    scalar negation
                        |
YACC rule #430 (p. 28)
YACC rule #310 (p. 22)
YACC rule #473 (p. 34)
                        |
YACC rule #429 (p. 28)
YACC rule #130 (p. 19)
YACC rule #473 (p. 34)
                        ;

 operand_381             :
YACC rule #382 (p. 24)
                        |
YACC rule #382 (p. 24)
YACC rule #804 (p. 38)
YACC rule #381 (p. 24)
YACC rule #466 (p. 33)
                        |
YACC rule #382 (p. 24)
YACC rule #823 (p. 40)
YACC rule #381 (p. 24)
YACC rule #466 (p. 33)
                        ;
 operand_A_382           :
YACC rule #383 (p. 24)
                        |
YACC rule #382 (p. 24)
YACC rule #421 (p. 27)
YACC rule #383 (p. 24)
                        ;
 operand_B_383           :
YACC rule #385 (p. 25)
                        |
```

```
YACC rule #385 (p. 25)
YACC rule #803 (p. 38)
YACC rule #383 (p. 24)
                        |
YACC rule #385 (p. 25)
YACC rule #822 (p. 40)
YACC rule #383 (p. 24)
                        ;
 operand_C_385           :
YACC rule #300 (p. 21)
                        |
YACC rule #817 (p. 39)
YACC rule #461 (p. 33)
    lerfu string as operand — classic math variable
                        |
YACC rule #428 (p. 28)
YACC rule #130 (p. 19)
YACC rule #473 (p. 34)
    quantifies a bridi — inverse of -MOI
                        |
YACC rule #427 (p. 28)
YACC rule #90 (p. 16)
YACC rule #473 (p. 34)
    quantifies a sumti — inverse of LI
                        |
YACC rule #431 (p. 29)
YACC rule #313 (p. 23)
YACC rule #473 (p. 34)
                        |
YACC rule #807 (p. 38)
YACC rule #381 (p. 24)
YACC rule #816 (p. 39)
YACC rule #385 (p. 25)
                        |
YACC rule #483 (p. 35)
YACC rule #381 (p. 24)
YACC rule #463 (p. 33)
                        ;


 _400 series constructs are mostly specific strings, some of which may
   also be used by the lexer; the lexer should not use any reference to
   terminals numbered less than _400, as they have grammars composed on
   non-deterministic strings of selma'o.  Some above _400 also are this
   way, so care should be taken; this is especially true for those that
   reference
YACC rule #32 (p. 12).

 anaphora_400            :
YACC rule #555 (p. 6)
                        |
YACC rule #555 (p. 6)
YACC rule #32 (p. 12)
                        |
```

```
YACC rule #817 (p. 39)
YACC rule #461 (p. 33)
                        ;

 cmevla_404               :
YACC rule #405 (p. 26)
                        |
YACC rule #405 (p. 26)
YACC rule #32 (p. 12)
                        ;
 cmevla_A_405             :
YACC rule #518 (p. 6)   pause
                        |
YACC rule #405 (p. 26)
YACC rule #518 (p. 6)   pause
 multiple CMEVLA are identified morphologically (by the lexer) - - separated
   consonant & pause
                        ;

 bridi_valsi_407         :
YACC rule #408 (p. 26)
                        |
YACC rule #408 (p. 26)
YACC rule #32 (p. 12)
                        ;
 bridi_valsi_A_408       :
YACC rule #509 (p. 6)
                        |
YACC rule #824 (p. 40)
                        |
YACC rule #543 (p. 6)
                        |
YACC rule #543 (p. 6)
YACC rule #593 (p. 7)
                        ;
 para_mark_410           :
YACC rule #584 (p. 7)
                        |
YACC rule #584 (p. 7)
YACC rule #32 (p. 12)
                        |
YACC rule #584 (p. 7)
YACC rule #410 (p. 26)
                        ;

 indicators_411          :
YACC rule #412 (p. 26)
                        |
YACC rule #535 (p. 6)
YACC rule #412 (p. 26)
                        ;
 indicators_A_412        :
YACC rule #413 (p. 27)
                        |
```

```
YACC rule #412 (p. 26)
YACC rule #413 (p. 27)
                        ;
 indicator_413           :
YACC rule #612 (p. 7)
                        |
YACC rule #515 (p. 6)
                        |
YACC rule #612 (p. 7)
YACC rule #581 (p. 7)
                        |
YACC rule #515 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #619 (p. 7)
                        |
YACC rule #524 (p. 6)
                        |
YACC rule #536 (p. 6)
                        ;
 DOI_415                 :
YACC rule #525 (p. 6)
                        |
YACC rule #416 (p. 27)
                        |
YACC rule #416 (p. 27)
YACC rule #525 (p. 6)
                        ;
 COI_416                 :
YACC rule #417 (p. 27)
                        |
YACC rule #416 (p. 27)
YACC rule #417 (p. 27)
                        ;
 COI_A_417               :
YACC rule #520 (p. 6)
                        |
YACC rule #520 (p. 6)
YACC rule #581 (p. 7)
                        ;

 JOIK_EK_421             :
YACC rule #802 (p. 38)
                        |
YACC rule #806 (p. 38)
                        |
YACC rule #806 (p. 38)
YACC rule #32 (p. 12)
                        ;
 JOIK_JEK_422            :
YACC rule #806 (p. 38)
                        |
YACC rule #806 (p. 38)
YACC rule #32 (p. 12)
```

```
                        |
YACC rule #805 (p. 38)
                        |
YACC rule #805 (p. 38)
YACC rule #32 (p. 12)
                        ;

 XI_424                  :
YACC rule #618 (p. 7)
                        |
YACC rule #618 (p. 7)
YACC rule #32 (p. 12)
                        ;
 NU_425                  :
YACC rule #426 (p. 28)
                        |
YACC rule #425 (p. 28)
YACC rule #422 (p. 27)
YACC rule #426 (p. 28)
                        ;
 NU_A_426                :
YACC rule #586 (p. 7)
                        |
YACC rule #586 (p. 7)
YACC rule #581 (p. 7)
                        |
YACC rule #586 (p. 7)
YACC rule #32 (p. 12)
                        |
YACC rule #586 (p. 7)
YACC rule #581 (p. 7)
YACC rule #32 (p. 12)
                        ;
 MOhE_427                :
YACC rule #664 (p. 8)
                        |
YACC rule #664 (p. 8)
YACC rule #32 (p. 12)
                        ;
 NIhE_428                :
YACC rule #666 (p. 8)
                        |
YACC rule #666 (p. 8)
YACC rule #32 (p. 12)
                        ;
 NAhU_429                :
YACC rule #665 (p. 8)
                        |
YACC rule #665 (p. 8)
YACC rule #32 (p. 12)
                        ;
 MAhO_430                :
YACC rule #662 (p. 8)
                        |
```

```
YACC rule #662 (p. 8)
YACC rule #32 (p. 12)
                        ;
 JOhI_431                :
YACC rule #657 (p. 8)
                        |
YACC rule #657 (p. 8)
YACC rule #32 (p. 12)
                        ;
 quote_arg_432           :
YACC rule #433 (p. 29)
                        |
YACC rule #433 (p. 29)
YACC rule #32 (p. 12)
                        ;
 quote_arg_A_433         :
YACC rule #434 (p. 29)
                        |
YACC rule #435 (p. 29)
                        |
YACC rule #436 (p. 29)
                        |
YACC rule #571 (p. 7)
YACC rule #0 (p. 9)
YACC rule #448 (p. 31)
                        ;

 The quoted material in the following three terminals must be identified by
   the lexer, but no additional lexer processing is needed.
 ZOI_quote_434           :
YACC rule #627 (p. 7)
YACC rule #698 (p. 8)
                                      pause
YACC rule #699 (p. 8)  pause
YACC rule #698 (p. 8)
                        ;

 “pause” is morphemic, represented by . The lexer assembles
YACC rule #699 (p. 8)
 ZO_quote_435            :
YACC rule #626 (p. 7)
YACC rule #698 (p. 8)
                        ;

 “word” may not be a compound; but it can be any valid Lojban selma'o value,
   including ZO, ZOI, SI, SA, SU. The preparser will not lex the word per its
   normal selma'o.
 LOhU_quote_436          :
YACC rule #569 (p. 7)
YACC rule #697 (p. 8)
YACC rule #565 (p. 7)
                        ;

 “words” may be any Lojban words, with no claim of grammaticality; the
```

preparser will not lex the individual words per their normal selma'o; used to quote ungrammatical Lojban, equivalent to the * or ? writing convention for such text.

The preparser needs one bit of sophistication for this rule. A quoted string should be able to contain other quoted strings – this is only a problem for a LOhU quote itself, since the LEhU clossing this quote would otherwise close the outer quotes, which is incorrect. For this purpose, we will cheat on the use of ZO in such a quote (since this is ungrammatical text, it is a sign ignored by the parser). Use ZO to mark any nested quotation LOhU. The preparser then will absorb it by the ZO rule, before testing for LOhU. This is obviously not the standard usage for ZO, which would otherwise cause the result to be a sumti. But, since the result will be part of an unparsed string anyway, it doesn't matter.

It may be seen that any of the ZO/ZOI/LOhU trio of quotation markers may contain the powerful metalinguistic erasers. Since these quotations

are not parsed internally, these operators are ignored within the quote. To erase a ZO, then, two SI's are needed after giving a quoted word of any type. ZOI takes four SI's, with the ENTIRE BODY OF THE QUOTE treated as a single “word” since it is one selma'o. Thus one for the quote body, two for the single word delimiters, and one for the ZOI. In LOhU, the entire body is treated as a single word, so three SI's can erase it.

All rule terminator names with “gap” in them are potentially elidable, where such elision does not cause an ambiguity. This is implemented through use of the YACC “error” token, which effectively recovers from an elision.

```
 FIhO_437                :
YACC rule #532 (p. 6)
                        |
YACC rule #532 (p. 6)
YACC rule #32 (p. 12)
                        ;
 PEhO_438                :
YACC rule #673 (p. 8)
                        |
YACC rule #673 (p. 8)
YACC rule #32 (p. 12)
                        ;
 BIhE_439                :
YACC rule #650 (p. 7)
                        |
YACC rule #650 (p. 7)
YACC rule #32 (p. 12)
                        ;
 SEI_440                 :
YACC rule #597 (p. 7)
                        |
YACC rule #597 (p. 7)
YACC rule #32 (p. 12)
```

```
                        ;
 FUhA_441                :
YACC rule #655 (p. 7)
                        |
YACC rule #655 (p. 7)
YACC rule #32 (p. 12)
                        ;
 BEI_442                 :
YACC rule #505 (p. 6)
                        |
YACC rule #505 (p. 6)
YACC rule #32 (p. 12)
                        ;
 CO_443                  :
YACC rule #519 (p. 6)
                        |
YACC rule #519 (p. 6)
YACC rule #32 (p. 12)
                        ;
 CEI_444                 :
YACC rule #516 (p. 6)
                        |
YACC rule #516 (p. 6)
YACC rule #32 (p. 12)
                        ;
 NA_445                  :
YACC rule #578 (p. 7)
                        |
YACC rule #578 (p. 7)
YACC rule #32 (p. 12)
                        ;
 BE_446                  :
YACC rule #504 (p. 6)
                        |
YACC rule #504 (p. 6)
YACC rule #32 (p. 12)
                        ;
 TUhE_447                :
YACC rule #610 (p. 7)
                        |
YACC rule #610 (p. 7)
YACC rule #32 (p. 12)
                        ;

 LIhU_gap_448            :
YACC rule #567 (p. 7)
                        |  error
                        ;

 gap_450                 :
YACC rule #556 (p. 6)
                        |
YACC rule #556 (p. 6)
YACC rule #32 (p. 12)
```

```
                        |  error
                        ;
 front_gap_451           :
YACC rule #521 (p. 6)
                        |
YACC rule #521 (p. 6)
YACC rule #32 (p. 12)
                        ;
 MEX_gap_452             :
YACC rule #658 (p. 8)
                        |
YACC rule #658 (p. 8)
YACC rule #32 (p. 12)
                        |  error
                        ;
 KEI_gap_453             :
YACC rule #552 (p. 6)
                        |
YACC rule #552 (p. 6)
YACC rule #32 (p. 12)
                        |  error
                        ;
 TUhU_gap_454            :
YACC rule #611 (p. 7)
                        |
YACC rule #611 (p. 7)
YACC rule #32 (p. 12)
                        |  error
                        ;
 VAU_gap_456             :
YACC rule #614 (p. 7)
                        |
YACC rule #614 (p. 7)
YACC rule #32 (p. 12)
                        |  error
                        ;

 redundant to attach a free modifier on the following
 DOhU_gap_457            :
YACC rule #526 (p. 6)
                        |  error
                        ;
 FEhU_gap_458            :
YACC rule #531 (p. 6)
                        |
YACC rule #531 (p. 6)
YACC rule #32 (p. 12)
                        |  error
                        ;
 SEhU_gap_459            :
YACC rule #598 (p. 7)
                        |  error
 a free modifier on a discursive should be somewhere within the discursive.
   See
```

```
YACC rule #440 (p. 30)
                        ;
 NUhU_gap_460            :
YACC rule #588 (p. 7)
                        |
YACC rule #588 (p. 7)
YACC rule #32 (p. 12)
                        |  error
                        ;

 BOI_gap_461             :
YACC rule #651 (p. 7)
                        |
YACC rule #651 (p. 7)
YACC rule #32 (p. 12)
                        |  error
                        ;
 sub_gap_462             :
YACC rule #651 (p. 7)
                        |  error
                        ;

 LUhU_gap_463            :
YACC rule #573 (p. 7)
                        |
YACC rule #573 (p. 7)
YACC rule #32 (p. 12)
                        |  error
                        ;

 GEhU_gap_464            :
YACC rule #538 (p. 6)
                        |
YACC rule #538 (p. 6)
YACC rule #32 (p. 12)
                        |  error
                        ;

 MEhU_gap_465            :
YACC rule #575 (p. 7)
                        |
YACC rule #575 (p. 7)
YACC rule #32 (p. 12)
                        |  error
                        ;

 KEhE_gap_466            :
YACC rule #550 (p. 6)
                        |
YACC rule #550 (p. 6)
YACC rule #32 (p. 12)
                        |  error
                        ;
```

```
 BEhO_gap_467            :
YACC rule #506 (p. 6)
                        |
YACC rule #506 (p. 6)
YACC rule #32 (p. 12)
                        |  error
                        ;

 TOI_gap_468            :
YACC rule #607 (p. 7)
                        |  error
                        ;

 KUhO_gap_469           :
YACC rule #557 (p. 6)
                        |
YACC rule #557 (p. 6)
YACC rule #32 (p. 12)
                        |  error
                        ;

 left_bracket_470       :
YACC rule #677 (p. 8)
                        |
YACC rule #677 (p. 8)
YACC rule #32 (p. 12)
                        ;
 right_bracket_gap_471  :
YACC rule #678 (p. 8)
                        |
YACC rule #678 (p. 8)
YACC rule #32 (p. 12)
                        |  error
                        ;
 LOhO_gap_472           :
YACC rule #568 (p. 7)
                        |
YACC rule #568 (p. 7)
YACC rule #32 (p. 12)
                        |  error
                        ;
 TEhU_gap_473           :
YACC rule #675 (p. 8)
                        |
YACC rule #675 (p. 8)
YACC rule #32 (p. 12)
                        |  error
                        ;
 right_br_no_free_474   :
YACC rule #678 (p. 8)
                        |  error
                        ;
```

```
 NUhA_475                 :
YACC rule #667 (p. 8)
                         |
YACC rule #667 (p. 8)
YACC rule #32 (p. 12)
                         ;
 MOI_476                  :
YACC rule #663 (p. 8)
                         |
YACC rule #663 (p. 8)
YACC rule #32 (p. 12)
                         ;
 ME_477                   :
YACC rule #574 (p. 7)
                         |
YACC rule #574 (p. 7)
YACC rule #32 (p. 12)
                         ;
 JAI_478                  :
YACC rule #547 (p. 6)
                         |
YACC rule #547 (p. 6)
YACC rule #32 (p. 12)
                         ;
 BO_479                   :
YACC rule #508 (p. 6)
                         |
YACC rule #508 (p. 6)
YACC rule #32 (p. 12)
                         ;
 SE_480                   :
YACC rule #596 (p. 7)
                         |
YACC rule #596 (p. 7)
YACC rule #32 (p. 12)
                         ;

 FA_481                   :
YACC rule #527 (p. 6)
                         |
YACC rule #527 (p. 6)
YACC rule #32 (p. 12)
                         ;
 NAhE_482                 :
YACC rule #583 (p. 7)
                         |
YACC rule #583 (p. 7)
YACC rule #32 (p. 12)
                         ;
 qualifier_483            :
YACC rule #561 (p. 6)
                         |
YACC rule #561 (p. 6)
YACC rule #32 (p. 12)
```

```
                        |
YACC rule #809 (p. 38)
                        ;
 NOI_484                 :
YACC rule #585 (p. 7)
                        |
YACC rule #585 (p. 7)
YACC rule #32 (p. 12)
                        ;
 GOI_485                 :
YACC rule #542 (p. 6)
                        |
YACC rule #542 (p. 6)
YACC rule #32 (p. 12)
                        ;
 subscript_486           :
YACC rule #424 (p. 28)
YACC rule #812 (p. 39)
YACC rule #462 (p. 33)
                        |
YACC rule #424 (p. 28)
YACC rule #470 (p. 34)
YACC rule #310 (p. 22)
YACC rule #474 (p. 34)
                        |
YACC rule #424 (p. 28)
YACC rule #817 (p. 39)
YACC rule #462 (p. 33)
                        ;
 ZIhE_487                :
YACC rule #625 (p. 7)
                        |
YACC rule #625 (p. 7)
YACC rule #32 (p. 12)
                        ;
 LE_488                  :
YACC rule #562 (p. 6)
                        |
YACC rule #562 (p. 6)
YACC rule #32 (p. 12)
                        ;
 LI_489                  :
YACC rule #566 (p. 7)
                        |
YACC rule #566 (p. 7)
YACC rule #32 (p. 12)
                        ;
 mod_head_490            :
YACC rule #491 (p. 36)
                        |
YACC rule #481 (p. 35)
                        ;

 tag_491                 :
```

```
YACC rule #815 (p. 39)
                        |
YACC rule #491 (p. 36)
YACC rule #422 (p. 27)
YACC rule #815 (p. 39)
                        ;
 ZOhU_492                :
YACC rule #628 (p. 7)
                        |
YACC rule #628 (p. 7)
YACC rule #32 (p. 12)
                        ;
 KE_493                  :
YACC rule #551 (p. 6)
                        |
YACC rule #551 (p. 6)
YACC rule #32 (p. 12)
                        ;
 PEhE_494                :
YACC rule #591 (p. 7)
                        |
YACC rule #591 (p. 7)
YACC rule #32 (p. 12)
                        ;
 CEhE_495                :
YACC rule #517 (p. 6)
                        |
YACC rule #517 (p. 6)
YACC rule #32 (p. 12)
                        ;
 NUhI_496                :
YACC rule #587 (p. 7)
                        |
YACC rule #587 (p. 7)
YACC rule #32 (p. 12)
                        ;
 VUhO_497                :
YACC rule #617 (p. 7)
                        |
YACC rule #617 (p. 7)
YACC rule #32 (p. 12)
                        ;
 SOI_498                 :
YACC rule #602 (p. 7)
                        |
YACC rule #602 (p. 7)
YACC rule #32 (p. 12)
                        ;
 LA_499                  :
YACC rule #558 (p. 6)
                        |
YACC rule #558 (p. 6)
YACC rule #32 (p. 12)
                        ;
```

```
 utterance_ordinal_801   :
YACC rule #905 (p. 40)
                        ;
 EK_802                  :
YACC rule #910 (p. 40)
                        |
YACC rule #910 (p. 40)
YACC rule #32 (p. 12)
                        ;
 EK_BO_803               :
YACC rule #915 (p. 41)
                        |
YACC rule #915 (p. 41)
YACC rule #32 (p. 12)
                        ;
 EK_KE_804               :
YACC rule #916 (p. 41)
                        |
YACC rule #916 (p. 41)
YACC rule #32 (p. 12)
                        ;
 JEK_805                 :
YACC rule #925 (p. 41)
                        ;
 JOIK_806                :
YACC rule #930 (p. 42)
                        ;
 GEK_807                 :
YACC rule #935 (p. 43)
                        |
YACC rule #935 (p. 43)
YACC rule #32 (p. 12)
                        ;
 GUhEK_808               :
YACC rule #940 (p. 43)
                        |
YACC rule #940 (p. 43)
YACC rule #32 (p. 12)
                        ;
 NAhE_BO_809             :
YACC rule #945 (p. 44)
                        |
YACC rule #945 (p. 44)
YACC rule #32 (p. 12)
                        ;
 NA_KU_810               :
YACC rule #950 (p. 44)
                        |
YACC rule #950 (p. 44)
YACC rule #32 (p. 12)
                        ;
 I_BO_811                :
YACC rule #955 (p. 44)
```

```
                        |
YACC rule #955 (p. 44)
YACC rule #32 (p. 12)
                        ;
 number_812              :
YACC rule #960 (p. 44)
                        ;
 GIhEK_BO_813            :
YACC rule #965 (p. 45)
                        |
YACC rule #965 (p. 45)
YACC rule #32 (p. 12)
                        ;
 GIhEK_KE_814            :
YACC rule #966 (p. 45)
                        |
YACC rule #966 (p. 45)
YACC rule #32 (p. 12)
                        ;
 tense_modal_815         :
YACC rule #970 (p. 45)
                        |
YACC rule #970 (p. 45)
YACC rule #32 (p. 12)
                        |
YACC rule #437 (p. 30)
YACC rule #130 (p. 19)
YACC rule #458 (p. 32)
                        ;
 GIK_816                 :
YACC rule #980 (p. 47)
                        |
YACC rule #980 (p. 47)
YACC rule #32 (p. 12)
                        ;

 lerfu_string_817        :
YACC rule #985 (p. 47)
                        ;
 GIhEK_818               :
YACC rule #990 (p. 47)
                        |
YACC rule #990 (p. 47)
YACC rule #32 (p. 12)
                        ;
 I_819                   :
YACC rule #995 (p. 48)
                        |
YACC rule #995 (p. 48)
YACC rule #32 (p. 12)
                        ;
 I_JEK_820               :
YACC rule #1000 (p. 48)
                        |
```

```
YACC rule #1000 (p. 48)
YACC rule #32 (p. 12)
                        ;
 JEK_BO_821              :
YACC rule #1005 (p. 48)
                        |
YACC rule #1005 (p. 48)
YACC rule #32 (p. 12)
                        ;
 JOIK_BO_822             :
YACC rule #1010 (p. 48)
                        |
YACC rule #1010 (p. 48)
YACC rule #32 (p. 12)
                        ;
 JOIK_KE_823             :
YACC rule #1015 (p. 48)
                        |
YACC rule #1015 (p. 48)
YACC rule #32 (p. 12)
                        ;
 PA_MOI_824              :
YACC rule #1025 (p. 49)
                        ;


 The following rules are used only in lexer processing.  They have been
   tested for ambiguity at various levels in the YACC grammar, but are in

   the recursive descent lexer in the current parser.  The lexer inserts
   the lexer tokens before the processed strings, but leaves the original
   tokens.
 lexer_A_905             :
YACC rule #701 (p. 8)
YACC rule #906 (p. 40)
                        ;
 utt_ordinal_root_906    :
YACC rule #986 (p. 47)
YACC rule #661 (p. 8)
                        |
YACC rule #961 (p. 44)
YACC rule #661 (p. 8)
                        ;

 lexer_B_910             :
YACC rule #702 (p. 8)
YACC rule #911 (p. 40)
                        ;
 EK_root_911             :
YACC rule #501 (p. 6)
                        |
YACC rule #596 (p. 7)
YACC rule #501 (p. 6)
                        |
```

```
YACC rule #578 (p. 7)
YACC rule #501 (p. 6)
                        |
YACC rule #501 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #596 (p. 7)
YACC rule #501 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #578 (p. 7)
YACC rule #501 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #578 (p. 7)
YACC rule #596 (p. 7)
YACC rule #501 (p. 6)
                        |
YACC rule #578 (p. 7)
YACC rule #596 (p. 7)
YACC rule #501 (p. 6)
YACC rule #581 (p. 7)
                        ;

 lexer_C_915             :
YACC rule #703 (p. 8)
YACC rule #911 (p. 40)
YACC rule #508 (p. 6)
                        |
YACC rule #703 (p. 8)
YACC rule #911 (p. 40)
YACC rule #971 (p. 45)
YACC rule #508 (p. 6)
                        ;

 lexer_D_916             :
YACC rule #704 (p. 8)
YACC rule #911 (p. 40)
YACC rule #551 (p. 6)
                        |
YACC rule #704 (p. 8)
YACC rule #911 (p. 40)
YACC rule #971 (p. 45)
YACC rule #551 (p. 6)
                        ;

 lexer_E_925             :
YACC rule #705 (p. 8)
YACC rule #926 (p. 41)
                        ;
 JEK_root_926            :
YACC rule #546 (p. 6)
                        |
YACC rule #546 (p. 6)
```

```
YACC rule #581 (p. 7)
                        |
YACC rule #578 (p. 7)
YACC rule #546 (p. 6)
                        |
YACC rule #578 (p. 7)
YACC rule #546 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #596 (p. 7)
YACC rule #546 (p. 6)
                        |
YACC rule #596 (p. 7)
YACC rule #546 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #578 (p. 7)
YACC rule #596 (p. 7)
YACC rule #546 (p. 6)
                        |
YACC rule #578 (p. 7)
YACC rule #596 (p. 7)
YACC rule #546 (p. 6)
YACC rule #581 (p. 7)
                        ;

 lexer_F_930             :
YACC rule #706 (p. 8)
YACC rule #931 (p. 42)
                        ;
 JOIK_root_931           :
YACC rule #548 (p. 6)
                        |
YACC rule #548 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #596 (p. 7)
YACC rule #548 (p. 6)
                        |
YACC rule #596 (p. 7)
YACC rule #548 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #932 (p. 42)
                        |
YACC rule #656 (p. 8)
YACC rule #932 (p. 42)
YACC rule #656 (p. 8)
                        ;
 interval_932            :
YACC rule #507 (p. 6)
                        |
YACC rule #507 (p. 6)
YACC rule #581 (p. 7)
```

```
                        |
YACC rule #596 (p. 7)
YACC rule #507 (p. 6)
                        |
YACC rule #596 (p. 7)
YACC rule #507 (p. 6)
YACC rule #581 (p. 7)
                        ;



 lexer_G_935             :
YACC rule #707 (p. 8)
YACC rule #537 (p. 6)
                        |
YACC rule #707 (p. 8)
YACC rule #596 (p. 7)
YACC rule #537 (p. 6)
                        |
YACC rule #707 (p. 8)
YACC rule #537 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #707 (p. 8)
YACC rule #596 (p. 7)
YACC rule #537 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #707 (p. 8)
YACC rule #971 (p. 45)
YACC rule #981 (p. 47)
                        |
YACC rule #707 (p. 8)
YACC rule #931 (p. 42)
YACC rule #539 (p. 6)
                        ;

 lexer_H_940             :
YACC rule #708 (p. 8)
YACC rule #544 (p. 6)
                        |
YACC rule #708 (p. 8)
YACC rule #596 (p. 7)
YACC rule #544 (p. 6)
                        |
YACC rule #708 (p. 8)
YACC rule #544 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #708 (p. 8)
YACC rule #596 (p. 7)
YACC rule #544 (p. 6)
YACC rule #581 (p. 7)
                        ;
```

```
 lexer_I_945             :
YACC rule #709 (p. 8)
YACC rule #583 (p. 7)
YACC rule #508 (p. 6)
                        ;

 lexer_J_950             :
YACC rule #710 (p. 8)
YACC rule #578 (p. 7)
YACC rule #556 (p. 6)
                        ;

 lexer_K_955             :
YACC rule #711 (p. 8)
YACC rule #956 (p. 44)
YACC rule #508 (p. 6)
                        |
YACC rule #711 (p. 8)
YACC rule #956 (p. 44)
YACC rule #971 (p. 45)
YACC rule #508 (p. 6)
                        ;
 I_root_956              :
YACC rule #545 (p. 6)
                        |
YACC rule #545 (p. 6)
YACC rule #957 (p. 44)
                        ;

 simple_JOIK_JEK_957     :
YACC rule #806 (p. 38)
                        |
YACC rule #805 (p. 38)
                        ;
                         no freemod in this version; cf.
YACC rule #422 (p. 27)
                         this reference to a version of JOIK and JEK
                           which already have the lexer tokens attached
                           prevents shift/reduce errors.  The problem is
                           resolved in a hard-coded parser implementation
                           which builds lexer_K, before lexer_S, before
                           lexer_E and lexer_F.


 lexer_L_960             :
YACC rule #712 (p. 8)
YACC rule #961 (p. 44)
                        ;

 number_root_961         :
YACC rule #672 (p. 8)
                        |
YACC rule #961 (p. 44)
```

```
YACC rule #672 (p. 8)
                        |
YACC rule #961 (p. 44)
YACC rule #987 (p. 47)
                        ;
 lexer_M_965             :
YACC rule #713 (p. 8)
YACC rule #991 (p. 47)
YACC rule #508 (p. 6)
                        |
YACC rule #713 (p. 8)
YACC rule #991 (p. 47)
YACC rule #971 (p. 45)
YACC rule #508 (p. 6)
                        ;

 lexer_N_966             :
YACC rule #714 (p. 8)
YACC rule #991 (p. 47)
YACC rule #551 (p. 6)
                        |
YACC rule #714 (p. 8)
YACC rule #991 (p. 47)
YACC rule #971 (p. 45)
YACC rule #551 (p. 6)
                        ;

 lexer_O_970             :
YACC rule #715 (p. 8)
YACC rule #972 (p. 45)
                        ;
 the following rule is a lexer version of non-terminal_815 for compounding
   PU/modals; it disallows the lexer picking out FIhO clauses, which would
   require it to have knowledge of the main parser grammar
 simple_tag_971          :
YACC rule #972 (p. 45)
                        |
YACC rule #971 (p. 45)
YACC rule #957 (p. 44)
YACC rule #972 (p. 45)
                        ;

 simple_tense_modal_972  :
YACC rule #973 (p. 45)
                        |
YACC rule #583 (p. 7)
YACC rule #973 (p. 45)
                        |
YACC rule #554 (p. 6)
                        |
YACC rule #522 (p. 6)
                        ;
 simple_tense_modal_A_973:
YACC rule #974 (p. 46)
```

```
                        |
YACC rule #974 (p. 46)
YACC rule #554 (p. 6)
                        |
YACC rule #977 (p. 46)
                        ;
 modal_974               :
YACC rule #975 (p. 46)
                        |
YACC rule #975 (p. 46)
YACC rule #581 (p. 7)
                        ;
 modal_A_975             :
YACC rule #502 (p. 6)
                        |
YACC rule #596 (p. 7)
YACC rule #502 (p. 6)
                        ;
 tense_A_977             :
YACC rule #978 (p. 46)
                        |
YACC rule #978 (p. 46)
YACC rule #554 (p. 6)
                        ;
 tense_B_978             :
YACC rule #979 (p. 46)
                        |
YACC rule #514 (p. 6)
                        |
YACC rule #979 (p. 46)
YACC rule #514 (p. 6)
                        ;
 specifies actuality/potentiality of the bridi


 puca'a = actually was
 baca'a = actually will be
 bapu'i = can and will have
 banu'o = can, but won't have yet
 canu'ojebapu'i = can, hasn't yet, but will
 tense_C_979             :
YACC rule #1030 (p. 49)
    time-only
    space defaults to time-space reference space

                        |
YACC rule #1040 (p. 50)
    can include time if specified with VIhA; otherwise time defaults to the
      time-space reference time

                        |
YACC rule #1030 (p. 49)
YACC rule #1040 (p. 50)
    time and space — If
```

```
YACC rule #1040 (p. 50) is marked with
   VIhA for space-time the tense may be self-contradictory
    interval prop before space_time is for time distribution
                          |
YACC rule #1040 (p. 50)
YACC rule #1030 (p. 49)
                          ;
 lexer_P_980               :
YACC rule #716 (p. 8)
YACC rule #981 (p. 47)
                          ;
 GIK_root_981              :
YACC rule #539 (p. 6)
                          |
YACC rule #539 (p. 6)
YACC rule #581 (p. 7)
                          ;
 lexer_Q_985               :
YACC rule #717 (p. 8)
YACC rule #986 (p. 47)
                          ;
 lerfu_string_root_986     :
YACC rule #987 (p. 47)
                          |
YACC rule #986 (p. 47)
YACC rule #987 (p. 47)
                          |
YACC rule #986 (p. 47)
YACC rule #672 (p. 8)
                          ;
 lerfu_word_987            :
YACC rule #513 (p. 6)
                          |
YACC rule #559 (p. 6)
YACC rule #987 (p. 47)
                          |
YACC rule #605 (p. 7)
YACC rule #986 (p. 47)
YACC rule #533 (p. 6)
                          ;

 lexer_R_990               :
YACC rule #718 (p. 8)
YACC rule #991 (p. 47)
                          ;

 GIhEK_root_991            :
YACC rule #541 (p. 6)
                          |
YACC rule #596 (p. 7)
YACC rule #541 (p. 6)
                          |
YACC rule #578 (p. 7)
YACC rule #541 (p. 6)
```

```
                        |
YACC rule #541 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #596 (p. 7)
YACC rule #541 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #578 (p. 7)
YACC rule #541 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #578 (p. 7)
YACC rule #596 (p. 7)
YACC rule #541 (p. 6)
                        |
YACC rule #578 (p. 7)
YACC rule #596 (p. 7)
YACC rule #541 (p. 6)
YACC rule #581 (p. 7)
                        ;

 lexer_S_995             :
YACC rule #719 (p. 8)
YACC rule #545 (p. 6)
                        ;
 lexer_T_1000            :
YACC rule #720 (p. 8)
YACC rule #545 (p. 6)
YACC rule #957 (p. 44)
                        ;

 lexer_U_1005            :
YACC rule #721 (p. 8)
YACC rule #926 (p. 41)
YACC rule #508 (p. 6)
                        |
YACC rule #721 (p. 8)
YACC rule #926 (p. 41)
YACC rule #971 (p. 45)
YACC rule #508 (p. 6)
                        ;
 lexer_V_1010            :
YACC rule #722 (p. 8)
YACC rule #931 (p. 42)
YACC rule #508 (p. 6)
                        |
YACC rule #722 (p. 8)
YACC rule #931 (p. 42)
YACC rule #971 (p. 45)
YACC rule #508 (p. 6)
                        ;
 lexer_W_1015            :
YACC rule #723 (p. 8)
```

```
YACC rule #931 (p. 42)
YACC rule #551 (p. 6)
                        |
YACC rule #723 (p. 8)
YACC rule #931 (p. 42)
YACC rule #971 (p. 45)
YACC rule #551 (p. 6)
                        ;
 lexer_Y_1025            :
YACC rule #725 (p. 8)
YACC rule #961 (p. 44)
YACC rule #663 (p. 8)
                        |
YACC rule #725 (p. 8)
YACC rule #986 (p. 47)
YACC rule #663 (p. 8)
                        ;


 time_1030               :
YACC rule #624 (p. 7)
                        |
YACC rule #624 (p. 7)
YACC rule #1031 (p. 49)
                        |
YACC rule #1031 (p. 49)
                        ;
 time_A_1031             :
YACC rule #1032 (p. 49)
                        |
YACC rule #1034 (p. 49)
                        |
YACC rule #1032 (p. 49)
YACC rule #1034 (p. 49)
                        ;
 time_B_1032             :
YACC rule #1033 (p. 49)
                        |
YACC rule #1032 (p. 49)
YACC rule #1033 (p. 49)
                        ;

 time_offset_1033        :
YACC rule #1035 (p. 50)
                        |
YACC rule #1035 (p. 50)
YACC rule #624 (p. 7)
                        ;

 time_interval_1034      :
YACC rule #622 (p. 7)
                        |
YACC rule #622 (p. 7)
YACC rule #1035 (p. 50)
```

```
                        |
YACC rule #1036 (p. 50)
                        |
YACC rule #622 (p. 7)
YACC rule #1036 (p. 50)
                        |
YACC rule #622 (p. 7)
YACC rule #1035 (p. 50)
YACC rule #1036 (p. 50)
                        ;
 time_direction_1035     :
YACC rule #592 (p. 7)
                        |
YACC rule #592 (p. 7)
YACC rule #581 (p. 7)
                        ;
 time_int_props_1036     :
YACC rule #1051 (p. 52)
                        |
YACC rule #1036 (p. 50)
YACC rule #1051 (p. 52)
                        ;

 space_1040              :
YACC rule #1042 (p. 50)
                        |
YACC rule #1041 (p. 50)
                        |
YACC rule #1042 (p. 50)
YACC rule #1041 (p. 50)
                        ;

 space_motion_1041       :
YACC rule #577 (p. 7)
YACC rule #1045 (p. 51)
                        ;
 space_A_1042            :
YACC rule #613 (p. 7)
                        |
YACC rule #613 (p. 7)
YACC rule #1043 (p. 50)
                        |
YACC rule #1043 (p. 50)
                        ;
 space_B_1043            :
YACC rule #1044 (p. 50)
                        |
YACC rule #1046 (p. 51)
                        |
YACC rule #1044 (p. 50)
YACC rule #1046 (p. 51)
                        ;
 space_C_1044            :
YACC rule #1045 (p. 51)
```

```
                          |
YACC rule #1044 (p. 50)
YACC rule #1045 (p. 51)
                          ;

 space_offset_1045       :
YACC rule #1048 (p. 51)
                          |
YACC rule #1048 (p. 51)
YACC rule #613 (p. 7)
                          ;

 space_intval_1046       :
YACC rule #1047 (p. 51)
                          |
YACC rule #1047 (p. 51)
YACC rule #1048 (p. 51)
                          |
YACC rule #1049 (p. 51)
                          |
YACC rule #1047 (p. 51)
YACC rule #1049 (p. 51)
                          |
YACC rule #1047 (p. 51)
YACC rule #1048 (p. 51)
YACC rule #1049 (p. 51)
                          ;

 space_intval_A_1047     :
YACC rule #615 (p. 7)
                          |
YACC rule #616 (p. 7)
                          |
YACC rule #615 (p. 7)
YACC rule #616 (p. 7)
                          ;
 space_direction_1048    :
YACC rule #528 (p. 6)
                          |
YACC rule #528 (p. 6)
YACC rule #581 (p. 7)
                          ;
 space_int_props_1049    :
YACC rule #1050 (p. 51)
                          |
YACC rule #1049 (p. 51)
YACC rule #1050 (p. 51)
                          ;
 space_int_props_A_1050  :
YACC rule #530 (p. 6)
YACC rule #1051 (p. 52)
                          ;

 This terminal gives an interval size in space-time (VEhA), and possibly a
```

dimensionality of the interval. The dimensionality may also be used

with the interval size left unspecified. When this terminal is used for t

spacetime origin, then barring any overriding VIhA, a VIhA here defines
the dimensionality of the space-time being discussed.

interval_property_1051 :
YACC rule #961 (p. 44)
YACC rule #594 (p. 7)
                        |
YACC rule #961 (p. 44)
YACC rule #594 (p. 7)
YACC rule #581 (p. 7)
                        |
YACC rule #604 (p. 7)
                        |
YACC rule #604 (p. 7)
YACC rule #581 (p. 7)
                        |
YACC rule #621 (p. 7)
                        |
YACC rule #621 (p. 7)
YACC rule #581 (p. 7)
                        ;

extensional/intensional interval parameters
These may be appended to any defined interval, or may stand in place of
either time or space tenses. If no other tense is present, this terminal

stands for the time-space interval parameter with an unspecified interval

roroi = always and everywhere

roroiku'avi = always here (ku'a = intersection)

puroroi = always in the past
paroi = once upon a time (somewhere)
paroiku'avi = once upon a time here

The following are “Lexer-only rules”, covered by steps 1-4 described
at the beginning. The grammar of these constructs is nonexistent,
except possibly in cases where they interact with each other. Even
there, however, the effects are semantic rather than grammatical. Where
it is believed possible that conflicts could exist, the grammar of these
constructs has been put in the above grammar, even though the
lexer/Preparser will actually prevent these from being passed thru to
the parse routine. (Otherwise we have to put unacceptably fancy code in

```
   the PreParser to determine just when these can be passed thru, and when
   they can't.)  Constructs in this category include quotes and indicators
   as defined above.  (The above grammar handles utterance scope
   (free_modifier) and clause scope (gap) applications of the latter,
   however, and indicators should be allowed to be absorbed into almost any
   word without changing its grammar.


YACC rule #601 (p. 7),
YACC rule #595 (p. 7), and
YACC rule #603 (p. 7) are metalinguistic erasers.

 token_1100              :
YACC rule #698 (p. 8)
                        |
YACC rule #503 (p. 6)
YACC rule #698 (p. 8)
                        |
YACC rule #699 (p. 8)
                        |
YACC rule #698 (p. 8)
YACC rule #511 (p. 6)
                        |
YACC rule #698 (p. 8)
YACC rule #524 (p. 6)
                        |
YACC rule #698 (p. 8)
YACC rule #536 (p. 6)
                        |
YACC rule #698 (p. 8)
YACC rule #535 (p. 6)
                        |
YACC rule #698 (p. 8)
YACC rule #612 (p. 7)
                        |
YACC rule #698 (p. 8)
YACC rule #612 (p. 7)
YACC rule #581 (p. 7)
                        |
YACC rule #698 (p. 8)
YACC rule #619 (p. 7)
                        |
YACC rule #698 (p. 8)
YACC rule #515 (p. 6)
                        |
YACC rule #698 (p. 8)
YACC rule #515 (p. 6)
YACC rule #581 (p. 7)
                        |
YACC rule #612 (p. 7)
YACC rule #581 (p. 7)
                        |
YACC rule #515 (p. 6)
YACC rule #581 (p. 7)
```

```
                        ;
 null_1101               :
YACC rule #698 (p. 8)
YACC rule #601 (p. 7)
                        |  possibly_unlexable_word  (PAUSE)
YACC rule #601 (p. 7)
                        |
YACC rule #20 (p. 11)
YACC rule #595 (p. 7)
                        |  possibly unlexable string  (PAUSE)
YACC rule #595 (p. 7)
                           erases back to the last individual token
                           I or NIhO or start of text, ignoring the
                           insides of ZOI, ZO, and LOhU/LEhU quotes.
                           Start of text is defined for SU below.
                        |
YACC rule #3 (p. 10)
YACC rule #603 (p. 7)
                        |  possibly unparsable text  (PAUSE)
YACC rule #603 (p. 7)
                           erases back to start of text which is the
                           beginning of a speaker's statement,
                           a parenthesis (TO/TOI), a LU/LIhU quote,
                           or a TUhE/TUhU utterance string.
                        ;
```

%%

1.2 EBNF grammar of Lojban

Lojban Machine Grammar, EBNF Version, Final Baseline

This EBNF document is explicitly dedicated to the public domain by its author, The Logical Language Group, Inc. Contact that organization at: 2904 Beau Lane, Fairfax VA 22031 USA 703-385-0273 (intl: +1 703 385 0273)

Explanation of notation: All rules have the form:

name $_{\text{number}}$ = bnf-expression

which means that the grammatical construct “name” is defined by “bnf-expression”.

1. Names in lower case are grammatical constructs.
2. Names in UPPER CASE are selma'o (lexeme) names, and are terminals.
3. Concatenation is expressed by juxtaposition with no operator symbol.
4. | represents alternation (choice).
5. [] represents an optional element.
6. & represents and/or. “A & B” is the same as “A | B | A B” but not “B A”. Furthermore, “A & B & C & D” permits one or more of A, B, C, and/or D, but only in that order.
7. ... represents optional repetition of the construct to the left. Left-grouping is implied; right-grouping is shown by explicit self-referential recursion with no “...”
8. () serves to indicate the grouping of the other operators. Otherwise, “...” binds closer than &, which binds closer than |.
9. # is shorthand for “[free ...]” , a construct which appears in many places.
10. // encloses an elidable terminator, which may be omitted (without change of meaning) if no grammatical ambiguity results.

text $_{0}$ =
    [NAI ...] [CMEVLA ... # | (indicators & free ...)] [joik-jek] text-1
text-1 $_{2}$ =
    [(I [jek | joik] [[stag] BO] #) ... | NIhO ... #] [paragraphs]
paragraphs $_{4}$ =
    paragraph [NIhO ... # paragraphs]
paragraph $_{10}$ =
    (statement | fragment) [I # [statement | fragment]] ...
statement $_{11}$ =
    statement-1 | prenex statement
statement-1 $_{12}$ =
    statement-2 [I joik-jek [statement-2]] ...
statement-2 $_{13}$ =
    statement-3 [I [jek | joik] [stag] BO # [statement-2]]
statement-3 $_{14}$ =
    sentence | [tag] TUhE # text-1 /TUhU#/
fragment $_{20}$ =
    ek # | gihek # | quantifier | NA # | terms /VAU#/ | prenex | relative-clauses | links | linkargs
prenex $_{30}$ =
    terms ZOhU #
sentence $_{40}$ =
    [terms [CU #]] bridi-tail
subsentence $_{41}$ =
    sentence | prenex subsentence
bridi-tail $_{50}$ =
    bridi-tail-1 [gihek [stag] KE # bridi-tail /KEhE#/ tail-terms]
bridi-tail-1 $_{51}$ =
    bridi-tail-2 [gihek # bridi-tail-2 tail-terms] ...
bridi-tail-2 $_{52}$ =
    bridi-tail-3 [gihek [stag] BO # bridi-tail-2 tail-terms]
bridi-tail-3 $_{53}$ =
    selbri tail-terms | gek-sentence
gek-sentence $_{54}$ =
    gek subsentence gik subsentence tail-terms | [tag] KE # gek-sentence /KEhE#/ | NA # gek-sentence
tail-terms $_{71}$ =
    [terms] /VAU#/
terms $_{80}$ =
    terms-1 ...
terms-1 $_{81}$ =
    terms-2 [PEhE # joik-jek terms-2] ...
terms-2 $_{82}$ =
    term [CEhE # term] ...
term $_{83}$ =
    sumti | (tag | FA #) (sumti | /KU#/) | termset | NA KU #
termset $_{85}$ =
    NUhI # gek terms /NUhU#/ gik terms /NUhU#/ | NUhI # terms /NUhU#/
sumti $_{90}$ =
    sumti-1 [VUhO # relative-clauses]
sumti-1 $_{91}$ =
    sumti-2 [(ek | joik) [stag] KE # sumti /KEhE#/]

sumti-2 92 =
sumti-3 [joik-ek sumti-3] ...
sumti-3 93 =
sumti-4 [(ek | joik) [stag] BO # sumti-3]
sumti-4 94 =
sumti-5 | gek sumti gik sumti-4
sumti-5 95 =
[quantifier] sumti-6 [relative-clauses] | quantifier selbri /KU#/ [relative-clauses]
sumti-6 97 =
(LAhE # | NAhE BO #) [relative-clauses] sumti /LUhU#/ | KOhA # | lerfu-string /BOI#/ | LA # [relative-clauses] CMEVLA ... # | (LA | LE) # sumti-tail /KU#/ | LI # mex /LOhO#/ | ZO any-word # | LU text /LIhU#/ | LOhU any-word ... LEhU # | ZOI any-word anything any-word #
sumti-tail 111 =
[sumti-6 [relative-clauses]] sumti-tail-1 | relative-clauses sumti-tail-1
sumti-tail-1 112 =
[quantifier] selbri [relative-clauses] | quantifier sumti
relative-clauses 121 =
relative-clause [ZIhE # relative-clause] ...
relative-clause 122 =
GOI # term /GEhU#/ | NOI # subsentence /KUhO#/
selbri 130 =
[tag] selbri-1
selbri-1 131 =
selbri-2 | NA # selbri
selbri-2 132 =
selbri-3 [CO # selbri-2]
selbri-3 133 =
selbri-4 ...
selbri-4 134 =
selbri-5 [joik-jek selbri-5 | joik [stag] KE # selbri-3 /KEhE#/] ...
selbri-5 135 =
selbri-6 [(jek | joik) [stag] BO # selbri-5]
selbri-6 136 =
tanru-unit [BO # selbri-6] | [NAhE #] guhek selbri gik selbri-6
tanru-unit 150 =
tanru-unit-1 [CEI # tanru-unit-1] ...
tanru-unit-1 151 =
tanru-unit-2 [linkargs]
tanru-unit-2 152 =
BRIVLA # | GOhA [RAhO] # | KE # selbri-3 /KEhE#/ | ME # sumti /MEhU#/ [MOI #] | (number | lerfu-string) MOI # | NUhA # mex-operator | SE # tanru-unit-2 | JAI # [tag] tanru-unit-2 | any-word (ZEI any-word) ... | NAhE # tanru-unit-2 | NU [NAI] # [joik-jek NU [NAI] #] ... subsentence /KEI#/
linkargs 160 =
BE # term [links] /BEhO#/
links 161 =
BEI # term [links]
quantifier 300 =
number /BOI#/ | VEI # mex /VEhO#/
mex 310 =
mex-1 [operator mex-1] ... | FUhA # rp-expression

mex-1 311 =
    mex-2 [BIhE # operator mex-1]
mex-2 312 =
    operand | [PEhO #] operator mex-2 ... /KUhE#/
rp-expression 330 =
    rp-operand rp-operand operator
rp-operand 332 =
    operand | rp-expression
operator 370 =
    operator-1 [joik-jek operator-1 | joik [stag] KE # operator /KEhE#/] ...
operator-1 371 =
    operator-2 | guhek operator-1 gik operator-2 | operator-2 (jek | joik) [stag] BO # operator-1
operator-2 372 =
    mex-operator | KE # operator /KEhE#/
mex-operator 374 =
    SE # mex-operator | NAhE # mex-operator | MAhO # mex /TEhU#/ | NAhU # selbri /TEhU#/ |
    VUhU #
operand 381 =
    operand-1 [(ek | joik) [stag] KE # operand /KEhE#/]
operand-1 382 =
    operand-2 [joik-ek operand-2] ...
operand-2 383 =
    operand-3 [(ek | joik) [stag] BO # operand-2]
operand-3 385 =
    quantifier | lerfu-string /BOI#/ | NIhE # selbri /TEhU#/ | MOhE # sumti /TEhU#/ | JOhI # mex-2 ...
    /TEhU#/ | gek operand gik operand-3 | (LAhE # | NAhE BO #) operand /LUhU#/
number 812 =
    PA [PA | lerfu-word] ...
lerfu-string 817 =
    lerfu-word [PA | lerfu-word] ...
lerfu-word 987 =
    BY | any-word BU | LAU lerfu-word | TEI lerfu-string FOI
ek 802 =
    [NA] [SE] A [NAI]
gihek 818 =
    [NA] [SE] GIhA [NAI]
jek 805 =
    [NA] [SE] JA [NAI]
joik 806 =
    [SE] JOI [NAI] | interval | GAhO interval GAhO
interval 932 =
    [SE] BIhI [NAI]
joik-ek 421 =
    joik # | ek #
joik-jek 422 =
    joik # | jek #
gek 807 =
    [SE] GA [NAI] # | joik GI # | stag gik
guhek 808 =
    [SE] GUhA [NAI] #
gik 816 =
    GI [NAI] #

tag 491 =
tense-modal [joik-jek tense-modal] ...
stag 971 =
simple-tense-modal [(jek | joik) simple-tense-modal] ...
tense-modal 815 =
simple-tense-modal # | FIhO # selbri /FEhU#/
simple-tense-modal 972 =
[NAhE] [SE] BAI [NAI] [KI] | [NAhE] (time [space] | space [time]) & CAhA [KI] | KI | CUhE
time 1030 =
ZI & time-offset ... & (ZEhA [PU [NAI]]) & interval-property ...
time-offset 1033 =
PU [NAI] [ZI]
space 1040 =
VA & space-offset ... & space-interval & (MOhI space-offset)
space-offset 1045 =
FAhA [NAI] [VA]
space-interval 1046 =
((VEhA & VIhA) [FAhA [NAI]]) & space-int-props
space-int-props 1049 =
(FEhE interval-property) ...
interval-property 1051 =
number ROI [NAI] | TAhE [NAI] | ZAhO [NAI]
free 32 =
SEI # [terms [CU #]] selbri /SEhU/ | SOI # sumti [sumti] /SEhU/ | vocative [relative-clauses] selbri [relative-clauses] /DOhU/ | vocative [relative-clauses] CMEVLA ... # [relative-clauses] /DOhU/ | vocative [sumti] /DOhU/ | (number | lerfu-string) MAI | TO text /TOI/ | XI # (number | lerfu-string) /BOI/ | XI # VEI # mex /VEhO/
vocative 415 =
(COI [NAI]) ... & DOI
indicators 411 =
[FUhE] indicator ...
indicator 413 =
(UI | CAI) [NAI] | Y | DAhO | FUhO

The following rules are non-formal:

word 1100 =
[BAhE] any-word [indicators]
any-word =
“any single word (no compound cmavo)”
anything =
“any text at all, whether Lojban or not”
null 1101 =
any-word SI | utterance SA | text SU

FAhO is a universal terminator and signals the end of parsable input.

Lojban Words Glossary

All definitions in this glossary are brief and unofficial. Only the published dictionary is a truly official reference for word definitions. These definitions are here simply as a quick reference.

le'u
: placeholder definition

lo'u
: placeholder definition

sa
: placeholder definition

si
: placeholder definition

su
: placeholder definition

zo
: placeholder definition

zoi
: placeholder definition

Lojban Words Index

General Index