

THE FOLLOWING ASSUMES: C-111 IS NORMALLY REVERSE-ORDERED. (LAST IN IS FIRST)

This applies the predicate to obj and returns the resulting value.
 First looks under TYPE properly and then slots; also check groups.
 Also check CAR of object.

checks to see if word could mark the end of a clause;
(DE CLAUSE-PUT (WORD)
(MEMO WORD & (CA PRO WHEN SUB))

check it as it is a creek in the input stream
(DE CLAUSE-BREAK-POINT (XOR))
(E2) XOR (PRC)

[illegible]

RECORDS IN THE CASE IS FILED, ALL IT IS NOT.
(DE IS FILED) (PARTIAL)
(NOT) (RECORD) (SELF-VALUE PARTIAL) (PARTIAL)

[illegible]

COF IF-FIND (L-800)
(DELT(VAR (VAR L-800) TEMP 810)
(SET) TEMP
(SUME
(LIST, LAMBDA (LIST VAR) (CONS, AND
(CONS (LIST, NOT (LIST, GET VAR, (EMBEDDED))
(COR L-800)))

<L-IND> is called just like LAMBDA: (L-IND <args> <>-expression).
 It builds a lambda-expression, which is applied in turn to the
 conceptualizations on the <LIST> from the most recently added to the
 last recently added (i.e., back to front), and returns the first for which
 the lambda-expression returns a non-nil value, also returns NIL.
 An embedding stack is added to the lambda-expression.

1. The following information is being furnished to you for your information only. It is not to be used for any other purpose.

* PRECEDES returns a non-NIL value iff C1 precedes C2 on the :C-LIST

* i.e., C1 was built before C2.

(DEF PRECEDES (C1 C2)

(AND (NOT (EQ C1 C2)) (MEMB C1 (MEMB C2 :C-LIST))))

* FOLLOWS returns a non-NIL value iff C1 follows C2 on the :C-LIST, i.e.

* C1 was built after C2 was.

(DEF FOLLOWS (C1 C2)

(AND (NOT (EQ C1 C2)) (MEMB C2 (MEMB C1 :C-LIST))))

* IM-FOLLOWS returns a non-NIL value iff C1 immediately follows C2 on the

* :C-LIST.

(DEF IM-FOLLOWS (C1 C2)

(EQ C2 (CADR (MEMB C1 :C-LIST))))

6.

Returns a non-NIL value iff X is a punctuation; applied at the lexical level;
possolent

6.

(DEFPROP BREAKPOINT

(LAMBDA (X) (MEMQ X '(PERIOD QUARK))))

EXPR)

*Check if a word is a verb but not a past participle

(DEF NOT-PPART (X)

(AND

(MEMB 'VERB (GET X 'POS:))

(MEMB 'PRESPART (GET X 'MORPH:))

Note that this solution is plausible from
a speech-processing standpoint, i.e.

present tense should default to PPART

* THESE DON'T SEARCH CORRECTLY ***** PLEASE NOTE *****

*Check if X is subject of Y (is sensitive to passive)

(DEF IS-SUBJ (X Y)

(COND

((MEMB 'PASSIVE-FLAG :FLAGS)

(FOLLOWS X Y))

(T (PRECEDES X Y))

*Check if X is object of Y

(DEF IS-OBJ (X Y)

(COND

((MEMB 'PASSIVE-FLAG :FLAGS)

(PRECEDES-R X Y))

(T (FOLLOWS-R X Y)))

*FOLLOWS-R first checks for an item which follows, then for an item which

*precedes and is unbound

(DEF FOLLOWS-R (X Y)

(OR

(FOLLOWS X Y)

(AND (NOT (GET X 'EMBEDDED)) (PRECEDES X Y)))

*PRECEDES-R checks for preceding unbound items

(DE: PRECEDES-R (X Y)

(AND: [NOT (GET X EMBEDDED) (PRECEDES X Y)]