File created: 12-Jun-90 10:02:15 **{DSK}<usr>local>lde>lispcore>library>MATCH.;2**

changes to: (VARS MATCHCOMS)

previous date: 10-Apr-84 21:34:35 {DSK}<usr>local>lde>lispcore>library>MATCH.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ MATCHCOMS
       ((FNS MAKEMATCH QMATCHSUBPAT QMATCHWM QMATCH$ QMATCH! QMATCH$= QMATCHELT1 QMATCHELT SIMPLEFN DOSIDE
             CHECKSETQ DOREPLACE DOREPLACE1)
        (FNS PATLEN $? ELT? SIMPLELT? ARB? NULLPAT? NILPAT CANMATCHNIL CANMATCHNILLIST REPLACEIN)
        (FNS EASYTORECOMPUTE GENSYML MAKESUBST DOSUBST DOSUBST1 SUBSTVAR BINDVAR SELFQUOTEABLE FINDIN0 FINDIN1
             DOWATCH PATNARGS)
        (FNS QNLEFT QNOT QNULL QNOT1 QNOTLESSPLENGTH QNTH QOR QPLUS QREPLACE MKAND QCAR QCDR QEQ QEQLENGTH
             QEQUAL QLAST QAPPLY* QLDIFF QFOR QLISTP QNCONC)
        (FNS PATERR PATHELP LOOKLIST VALUELOOKUP LOOK)
        (FNS MKAND2 CHECKSLISTP EQUALUNCROP)
        (FNS PATPARSE PATPARSE1 PATUNPACKINFIX1 PARSEDEFAULT VARCHECK PATUNPACK PATUNPACKINFIX PATGETFNNAME
             PATGETEXPR PATPARSEAT MAKE!PAT MAKESUBPAT NEGATEPAT PACKLDIFF)
        (VARS PATCHARS PATTERNINFIXES PATTERNINFIXES1 PATTERNREPLACEOPRS PATTERNITEMS NEVERNILFUNCTIONS
              PATNONNILFUNCTIONS [PATTERNCHARRAY (MAKEBITTABLE (NCONC (MAPCAR PATCHARS 'CAAR)
                                                                      (MAPCAR PATTERNITEMS 'CAR]
              PATGENSYMVARS
              (PATVARDEFAULT '=)
              MAXCDDDDRS
              (PATCHECKLENGTH T)
              (PATLISTPCHECK (EQ 'VAX (SYSTEMTYPE)))
              (PATVARSMIGHTBENIL T))
        (VARS PATCHARS PATTERNINFIXES PATTERNINFIXES1 PATTERNREPLACEOPRS PATTERNITEMS NEVERNILFUNCTIONS
              PATNONNILFUNCTIONS SIMPLE.PREDICATES [PATTERNCHARRAY (MAKEBITTABLE (NCONC (MAPCAR PATCHARS
                                                                                               'CAAR)
                                                                                        (MAPCAR PATTERNITEMS
                                                                                               'CAR]
              PATGENSYMVARS)
        (P (OR (BOUNDP 'MATCHSTATS)
               (SETQ MATCHSTATS)))
        (VARS PATVARDEFAULT MAXCDDDDRS (PATCHECKLENGTH T)
              (PATLISTPCHECK NIL)
              (PATVARSMIGHTBENIL T))
        (BLOCKS * MATCHBLOCKS)))

(DEFINEQ

(MAKEMATCH
  [LAMBDA (MATCHEXPRESSION PATTERN)                                (* lmm "22-NOV-82 12:08")
    (PROG ((LOCALDECLARATION (GETLOCALDEC EXPR FAULTFN))
           %#LIST %#LISTUSED BOUNDVARS BOUNDVALS CHECKLENGTH LISTPCHECK VARDEFAULT PATVARSNIL (GENSYMVARLIST
                                                                                               PATGENSYMVARS)
           CONSTRUCT POSTPONEDRPLACS POSTPONEDSETQS (LASTEFFECTCANBENIL T)
           MUSTRETURN WMLST WATCHPOSTPONELST SUBLIST PAT MATCHEFFECTS VAR X INASOME)

          (* POSTPONEDSETQS and POSTPONEDRPLACS are used to collect postponed side effects -
          LASTEFFECTCANBENIL is a flag which should be set whenever a SETQ is postponed for determining whether the extra T
          at the end is necessary -
          BOUNDVARS and BOUNDVALS will be list of bindings that need to be done -
          MUSTRETURN will be the * expression, if any)

          (* CHECKINGLENGTH is the flag whether the length should be checked
          (used for example in (-- %'A & &) already done the NLEFT which implicitly checks) -
          -
          INASOME is a stack, car of which is a that says that we are, at this level, after a --
          type pattern, so that if another -- is encountered, just reset INASOME to the match expression for what comes after the
          second --; this is so (-- A -- B --) will generate (MEMB %'B (MEMB %'A X)) instead of
          (SOME X (F/L (Z) (Z%:1='A AND %'B MEMB |Z::1|))) -
          WMLST is a stack used by *GLITCH for remembering when a !
          (SUBPAT --) is encountered to expand it, but remember the tail after the !SUBPAT and return
          (by RPLAC'ing into the corresponding entry in WMLIST) the expression for "WHAT MATCHED" -
          SUBLIST is the list where substitutions in the final pattern are collected)

          (* WATCHPOSTPONELST is a list of those vars which, when a POSTPONE involving them is encountered, the
          corresponding entry in WATCHPOSTPONELST should be rplac'ed ;
          used to tell whether SOME variables should be local or global)

          (SETQ CHECKLENGTH (VALUELOOKUP 'PATCHECKLENGTH))
          (SETQ LISTPCHECK (VALUELOOKUP 'PATLISTPCHECK))
          (SETQ VARDEFAULT (VALUELOOKUP 'PATVARDEFAULT))
          (SETQ PATVARSNIL (VALUELOOKUP 'PATVARSMIGHTBENIL))     (* Look up global variables, checking the local declaration)
          (SETQ CLISPCHANGE T)
          (* Tell DWIM that if the match fails not to try to parse it as CLISP)

          [COND
             (PATTERN                                               (* Old way of calling MAKEMATCH)
                    (SETQ MATCHEXPRESSION (LIST 'match MATCHEXPRESSION 'with PATTERN]
          (SELECTQ (CAR MATCHEXPRESSION)
               ((match MATCH)
                   (DWIMIFY0? (CDR MATCHEXPRESSION)
                          MATCHEXPRESSION
                          (CDR MATCHEXPRESSION)
                          T T FAULTFN)
                   [SELECTQ (CADDR MATCHEXPRESSION)
                        ((with WITH))
                        (COND
                           ((FIXSPELL (CADDR MATCHEXPRESSION)
                                   70
                                   '(WITH)
                                   T
                                   (CDDR MATCHEXPRESSION)))
                           ((AND (LISTP (CADDR MATCHEXPRESSION))
                                 (FIXSPELL1 (CADDR MATCHEXPRESSION)
                                        (CONS 'with (CADDR MATCHEXPRESSION))
                                        NIL T))
                            (/ATTACH 'with (CDDR MATCHEXPRESSION)))
                           (T (PATERR 'NOWITH (CDDR MATCHEXPRESSION]
                   [SELECTQ (CAR (CDDDDR MATCHEXPRESSION))
                        ((NIL -> =>))
                        (PATERR "Expression after pattern not preceded by => or ->" (CAR (CDDDDR MATCHEXPRESSION
                                                                                                          ])
               (HELP "Bad arg to MAKEMATCH"))                       (* Make sure expression is in right form)
          [SETQ PAT (PROG ((TOPPAT (CADDDR MATCHEXPRESSION)))
                          (RETURN (PATPARSE TOPPAT]                 (* Parse the pattern into internal format)
          [SETQ CONSTRUCT (AND (CDDDDR MATCHEXPRESSION)
                               (PROG ((VARS (APPEND %#LIST VARS)))
                                     (DWIMIFY0? (CDR (CDDDDR MATCHEXPRESSION))
                                            MATCHEXPRESSION T NIL NIL FAULTFN)
                                     (RETURN (CDR (CDDDDR MATCHEXPRESSION]
                                                                    (* Get any expression after => or ->)
          [SETQ VAR (COND
                             ((EASYTORECOMPUTE (CADR MATCHEXPRESSION))
                              (CADR MATCHEXPRESSION))
                             (T (SUBSTVAR (CADR MATCHEXPRESSION]
          (SETQ X (QMATCHSUBPAT VAR PAT))
          (SETQ SUBLIST (DREVERSE SUBLIST))
          [AND CONSTRUCT (EQ (CAR (CDDDDR MATCHEXPRESSION))
                             '->)
               (SETQ CONSTRUCT (LIST (LIST 'TOPREPLACE VAR (MKPROGN CONSTRUCT]
          [SETQ MATCHEFFECTS (NCONC (DREVERSE POSTPONEDSETQS)
                                    (DREVERSE POSTPONEDRPLACS)
                                    (COND
                                       (CONSTRUCT)
                                       (MUSTRETURN (LIST MUSTRETURN))
                                       ((AND LASTEFFECTCANBENIL (NULL POSTPONEDRPLACS))
                                        (LIST T]
          [SETQ X (DOSUBST (COND
                               [MATCHEFFECTS `(COND
                                                   (%, X %,@ MATCHEFFECTS]
                               (T X]
          (RETURN (COND
                     (BOUNDVARS `([LAMBDA %, BOUNDVARS %, X]
                                  %,@ BOUNDVALS))
                     (T X])

(QMATCHSUBPAT
  [LAMBDA (VAR PATELT NOLISTPCHECK)                                 (* lmm "10-AUG-78 15:47")
    (PROG ((CHECKINGLENGTH T)
           (INASOME (CONS NIL INASOME)))                            (* Rebind INASOME since this is on a different level;
                                                                    also CHECKINGLENGTH)
          (RETURN (COND
                     ((AND LISTPCHECK (NOT NOLISTPCHECK))
                      (MKAND (QLISTP VAR)
                             (QMATCHWM VAR PATELT)))
                     (T (QMATCHWM VAR PATELT])

(QMATCHWM
  [LAMBDA (VAR PAT FN)                                              (* lmm "10-AUG-78 15:47")

          (* Creates an expression which will return non-NIL if and only if the value of the VAR expression will match the parsed
          pattern PAT, and the expression generated by applying DOSIDE to
          (the expression giving What-Matched the first pattern element of PAT) and FN if FN is non-NIL -
          is non-nil as well.)

    (PROG (TEM1 TEM2)
          [COND
```

```
             ((NULL PAT)
              (RETURN (OR (NOT CHECKLENGTH)
                          (NOT CHECKINGLENGTH)
                          (QNULL VAR]
          [COND
             ((NLISTP (CAR PAT))                                         (* The only NLISTP patterns are &, $, --, NIL, T, strings and
                                                                         numbers)
              (RETURN (SELECTQ (CAR PAT)
                          (($ --)
                               (QMATCH$ VAR PAT FN))
                           (QMATCHELT1 VAR PAT FN]
          (SELECTQ (CAAR PAT)
               ((= == %' SUBPAT ~ *ANY*)

          (* For now, ~'s can only refer to = == %' and subpats %, i.e. elementary patterns)

                   (RETURN (QMATCHELT1 VAR PAT FN)))
               (! (RETURN (QMATCH! VAR PAT FN)))
               ($= (RETURN (QMATCH$= VAR PAT FN)))
               (@ (COND
                     [(ELT? (CDDAR PAT))
                      (COND
                         [(AND (OR (NEQ (CAR INASOME)
                                        'FASTINASOME)
                                   (NEQ (CDDAR PAT)
                                        '&))
                               (SIMPLEFN (CADAR PAT)))

          (* Put simple tests first %, unless it is just &@SIMPLEFN, in which case we want to go thru QMATCHELT1 so that the
          FASTINASOME will catch the &@FN; for example, in ($ &@LISTP %' A $) find a list and look for A after it, rather than find a
          list followed by A)

                          (RETURN (MKAND (QAPPLY* (CADAR PAT)
                                                  (QCAR VAR))
                                         (QMATCHWM VAR (CONS (CDDAR PAT)
                                                             (CDR PAT))
                                               FN]
                         (T (RETURN (QMATCHELT1 VAR PAT FN]
                     [(CDR PAT)                                          (* SEGMENT@FN followed by something)
                      (COND
                         [(AND (NULL FN)
                               ($? (CDDAR PAT))
                               (ARB? (CADR PAT)
                                     T))                                 (* $@FN followed by $ or $@)
                          [SETQ TEM1 (QFOR 'OLD (SETQ TEM1 (GENSYML))
                                           VAR
                                           (MKAND (QAPPLY* (CADAR PAT)
                                                          (QLDIFF VAR TEM1))
                                                  (QMATCHWM TEM1 (CDR PAT)))
                                           T
                                           (CANMATCHNILLIST (CDR PAT]
                          (COND
                             ((CAR INASOME)
                              (FRPLACA INASOME TEM1)
                              (RETURN T))
                             (T (RETURN TEM1]
                         (T

          (* segment@FN followed by more pattern -
          cannot assume that the INASOME check is legit since ($ (%' A $ %' B) @FOO $) check the B MUST repeatedly be checked
          for)

                            (RETURN (PROG ((INASOME (CONS NIL INASOME)))
                                          (RETURN (QMATCHWM VAR
                                                         [LIST (CDDAR PAT)
                                                               (CONS '@ (CONS (QAPPLY* (CADAR PAT)
                                                                                      (QLDIFF VAR '@))
                                                                              (MAKE!PAT (MAKESUBPAT
                                                                                             (CDR PAT]
                                                         FN]
                     (T (GO OTHER))))
               (GO OTHER))
      OTHER
          (RETURN (QMATCHWM VAR (CONS (CDDAR PAT)
                                      (CDR PAT))
                         (CONS (CAR PAT)
                               FN])

(QMATCH$
  [LAMBDA (VAR PAT FN)                                                   (* lmm "10-AUG-78 15:47")
    (PROG (TEM1 TEM2 ZLENFLG (SKIPEDLEN 0)
                TAIL)
          (RETURN (COND
                     ((NULL (CDR PAT))                                   (* Pattern ends in $ -
                                                                         What matched is the whole thing)
                      (DOSIDE FN VAR))
```

```
           ((CAR INASOME)

(* We are within a tail which began with -- or $; thus, we should not return the match here but instead, smash car of
INASOME to the match expression here and return T -
since there is no point in checking this match expression repeatedly)

            (COND
               ((LISTP (CAR INASOME))
                (PATHELP "INASOME mismatch")))
            [DOWATCH (FRPLACA INASOME (PROG ((INASOME (CONS NIL INASOME)))
                                            (RETURN (QMATCHWM VAR PAT FN]

(* Want to check for postponed variables -
Now just return T, and let the call that rebound INASOME pick up the expression)

            T)
           ((ARB? (CADR PAT))
            (PATERR "Two $ or -- patterns in a row, ambiguous")
                                                       (* Must mean the second is LAST)
            )
           [[AND (NULL FN)
                 (PROGN [SETQ TAIL (SOME (CDR PAT)
                                         (FUNCTION (LAMBDA (ELT)
                                                     (COND
                                                        ((EQ ELT '&)
                                                         (SETQ SKIPEDLEN (QPLUS 1 SKIPEDLEN))
                                                         NIL)
                                                        ((EQ (CAR ELT)
                                                             '$=)
                                                         (SETQ SKIPEDLEN (QPLUS SKIPEDLEN
                                                                                (CDR ELT)))
                                                         NIL)
                                                        (T]

(* Returns to the first TAIL of PAT which doesn't begin with a $i or a $$foo -
Sets the variable "LEN" to the total length of things skipped over)

                        (NOT (ZEROP SKIPEDLEN]         (* Special check here, since might have
                                                       (|...| -- $4) or not need any QNLEFT's)
            (COND
               ((OR (NULL TAIL)
                    (NULLPAT? TAIL))
                (QNOTLESSPLENGTH VAR SKIPEDLEN))
               ((NUMBERP SKIPEDLEN)                    (* Change ($ $4 |...|) to ($4 $ |...|))
                (QMATCHWM (QNTH VAR (ADD1 SKIPEDLEN))
                       (CONS (CAR PAT)
                             TAIL)))
               (T                                      (* same here; only the NTH expression is NOT to be substituted)
                  (QMATCHWM [QCDR (QCDR (SETQ TEM1 (SUBSTVAR (QNTH VAR SKIPEDLEN]
                         (CONS (CAR PAT)
                               TAIL]
           [[NILPAT (SETQ TAIL (SOME (CDR PAT)
                                     (FUNCTION (LAMBDA (ELT TEM)
                                                 (COND
                                                    ((NULL (SETQ TEM (PATLEN ELT)))
                                                     T)
                                                    ((ZEROP TEM)
                                                     (SETQ ZLENFLG T)
                                                     NIL)
                                                    (T (SETQ SKIPEDLEN (QPLUS SKIPEDLEN TEM))
                                                       NIL]

(* Scans PAT until a pattern element which matches an arbitrary length segment is hit -
Adds the length skipped to the variable SKIPEDLEN; and sets ZLENFLG if finds any of zero length)
                                                       (* Skipping over any arbitrary patterns
                                                       (though might have setqs in them) check if ends in NIL)
            (PROG (CHECKINGLENGTH)

(* If pat ends in (|...| -- & & &) then just match (NLEFT var 3) against & & &;
CHECKINGLENGTH = NIL will keep a (NULL (CDDDR x)) check away)

                  [SETQ TEM1 (COND
                                [[OR (ZEROP SKIPEDLEN)
                                     (AND (EQ (CAR (LISTP VAR))
                                              'CDR)
                                          (NOT (ELT? (CADR PAT)))
                                          (REPLACEIN (CADR PAT]
                                                       (* Check |var::-skipedlen|)
                                 (QCDR (SUBSTVAR (QNLEFT (COND
                                                            ((EQ (CAR (LISTP VAR))
                                                                 'CDR)
                                                             (CADR VAR))
                                                            (T VAR))
                                                       (QPLUS SKIPEDLEN 1)
                                                       NIL ZLENFLG]
                                (T (SUBSTVAR (QNLEFT VAR SKIPEDLEN NIL ZLENFLG]
                  (RETURN (MKAND (OR (NOT (CANMATCHNILLIST (CDR PAT)))
                                                 TEM1)
                                            (MKAND (QMATCHWM TEM1 (CDR PAT))
                                                   (OR (NULL FN)
                                                       (DOSIDE FN (QLDIFF VAR TEM1]
                     [[AND (NULL FN)
                           (EQ TAIL (CDDR PAT))
                           (EQ SKIPEDLEN 1)
                           (NULLPAT? TAIL)
                           (EQ (CAADR PAT)
                               'SUBPAT)
                           (OR (EQ (CAR PAT)
                                   '$)
                               (EVERY (CDDR (CADR PAT))
                                      (FUNCTION ARB?)))
                           [COND
                              [(NLISTP (CADR (CADR PAT)))
                               (NOT (FMEMB (CADR (CADR PAT))
                                           '(& $ --]
                              (T (FMEMB (CAR (CADR (CADR PAT)))
                                        '(= == %']
                           (FMEMB [CAR (SETQ TEM1 (QMATCHELT 'DUMMY (CADR (CADR PAT]
                                  '(EQ EQUAL EQP STREQUAL]      (* PAT%: (-- (SUBPAT EQTYPE? ARB?) --))
                      (PROG [TEM2 (VAR (LIST (SELECTQ (CAR TEM1)
                                                   (EQ (LOOK 'ASSOC VAR))
                                                   'SASSOC)
                                              (CADDR TEM1)
                                              VAR))
                                  (PAT (CONS '& (CDDR (CADR PAT]
                            (RETURN (QMATCHSUBPAT (SUBSTVAR VAR)
                                            PAT T]
                     (T (PROG ({OLD} {FINALLY}EXPR {UNTIL}EXPR {ON}VAR [INASOME (FRPLACA INASOME
                                                                                        (COND
                                                                                           ((CAR INASOME)
                                                                                            (PATHELP "INASOME
                                                                                                   mismatch"))
                                                                                           ((EQ (CAR PAT)
                                                                                                '$)
                                                                                            'FASTINASOME)
                                                                                           (T 'INASOME]
                                      (WATCHPOSTPONELST (CONS (SETQ TEM1 (GENSYML))
                                                              WATCHPOSTPONELST)))

          (* WATCHPOSTPONELST is reset so that postponed uses of it can be detected;
          needed to set {OLD})

                              (COND
                                 ((AND (EQ (CAR (LISTP VAR))
                                           'CDR)
                                       [for X in (CDR PAT) do (COND
                                                                 ((ELT? X)
                                                                  (RETURN))
                                                                 ((REPLACEIN X)
                                                                  (RETURN T]
                                       (SETQ {ON}VAR (CADR VAR)))
                                   (SETQ TEM2 (QCDR TEM1)))
                                 (T (SETQ {ON}VAR VAR)
                                    (SETQ TEM2 TEM1)))
                              (SETQ {UNTIL}EXPR (QMATCHWM TEM2 (CDR PAT)))
                              [SETQ {FINALLY}EXPR (COND
                                                      [(EQ {UNTIL}EXPR T)
                                                       (SELECTQ (CAR INASOME)
                                                           ((INASOME FASTINASOME NIL)
                                                                (PATHELP "bad pattern tail"))
                                                           (PROGN (SETQ {UNTIL}EXPR (CAR INASOME))
                                                                  (OR (NULL FN)
                                                                      (DOSIDE FN (QLDIFF VAR TEM2]
                                                      (T (MKAND (DOSIDE FN (QLDIFF VAR TEM2))
                                                                (OR [NEQ (FMEMB (CAR INASOME)
                                                                                '(INASOME FASTINASOME NIL]
                                                                    (CAR INASOME]
                              (SETQ {OLD} (EQ (CAR WATCHPOSTPONELST)
                                              'FOUND))
                              (RETURN (QFOR {OLD} TEM1 {ON}VAR {UNTIL}EXPR {FINALLY}EXPR (CANMATCHNILLIST
                                                                                             (CDR PAT])

(QMATCH!
  [LAMBDA (VAR PAT FN)                                              (* lmm "10-AUG-78 15:47")
    (PROG (TEM1)
          (RETURN
           (COND
              ((NILPAT (CDR PAT))
               (MKAND [COND
                           ((EQ (CADAR PAT)
                                'SUBPAT)                            (* This isn't really a subpat and so don't rebind
                                                                    CHECKINGLENGTH etc as in QMATCHSUBPAT)

                            (QMATCHWM VAR (CDDAR PAT)))
```

```
                             (T (QMATCHELT VAR (CDAR PAT]
                         (DOSIDE FN VAR)))
                ((NLISTP (CAR PAT))
                 (PATERR "Invalid '!'" PAT))
                (T
                 (SELECTQ (CADAR PAT)
                      (=                                                 (* != X -
                                                                         Go down VAR and X simultaneously, looking for EQUAL
                                                                         subelements)
                         [PROG ((TEMVAR (BINDVAR (GENSYML)))
                                 (TAILVAR (BINDVAR (GENSYML)))
                                AFTEREXP)
                                [SETQ AFTEREXP (MKAND (DOSIDE FN (QLDIFF VAR TAILVAR))
                                                        (QMATCHWM TAILVAR (CDR PAT]
                                (RETURN (SUBPAIR '(TAILVAR VAR TEMVAR ONVAR FINALLY)
                                                [LIST TAILVAR VAR TEMVAR (CDDAR PAT)
                                                       (COND
                                                          [(EQ AFTEREXP T)
                                                           (QOR (LIST (QNULL TEMVAR)
                                                                        (QEQUAL TEMVAR TAILVAR]
                                                          ((NOT (CANMATCHNILLIST (CDR PAT)))
                                                           (MKAND (QNULL TEMVAR)
                                                                   AFTEREXP))
                                                          (T (MKAND (QOR (LIST (QNULL TEMVAR)
                                                                                   (QEQUAL TEMVAR TAILVAR)))
                                                                     AFTEREXP]
                                                '(PROG NIL
                                                        (SETQ TEMVAR ONVAR)
                                                        (SETQ TAILVAR VAR)
                                                   $$LP
                                                        (COND
                                                           ((NLISTP TEMVAR)
                                                            (RETURN FINALLY))
                                                           ([OR (NLISTP TAILVAR)
                                                                 (NOT (EQUAL (CAR TEMVAR)
                                                                              (CAR TAILVAR]
                                                            (RETURN)))
                                                        (SETQ TAILVAR (CDR TAILVAR))
                                                        (SETQ TEMVAR (CDR TEMVAR))
                                                        (GO $$LP])
                      (== [COND
                             [(NULLPAT? (CDR PAT))
                              (PROG ((CHECKLENGTH T))
                                     (RETURN (QMATCHWM VAR (LIST (CAR PAT))
                                                     FN]
                             (T (PATERR '!AT (CDAR PAT])
                      (%' (COND
                             [[OR (NLISTP (CDDAR PAT))
                                   (CDR (LAST (CDDAR PAT]
                              (COND
                                 [(NULLPAT? (CDR PAT))
                                  (PROG ((CHECKLENGTH T))
                                         (RETURN (QMATCHWM VAR (LIST (CAR PAT))
                                                         FN]
                                 (T (PATERR '!AT (CDAR PAT]
                             (T (QMATCHWM VAR (CONS [CONS '! (CONS 'SUBPAT (MAPCAR (CDDAR PAT)
                                                                                          (FUNCTION (LAMBDA (X)
                                                                                                        (CONS '%' X]
                                                         (CDR PAT))
                                        FN))))
                      (SUBPAT                                            (* Use the *GLITCH kludge to get the whatmatched of the rest of
                              the thing)
                              [COND
                                 [(NULL FN)
                                  (QMATCHWM VAR (APPEND (CDDAR PAT)
                                                          (CDR PAT]
                                 (T
                                  (PROG ((WMLST (CONS NIL WMLST)))
                                         (RETURN
                                          (MKAND [QMATCHWM
                                                    VAR
                                                     (APPEND (CDDAR PAT)
                                                             (LIST (CONS '*GLITCH (CONS WMLST
                                                                                          (MAKE!PAT (MAKESUBPAT
                                                                                                       (CDR PAT]
                                                  (DOSIDE FN (QLDIFF VAR (CAR WMLST])
                      (PATERR "Invalid use of ! in pattern" (CADAR PAT])

(QMATCH$=
  [LAMBDA (VAR PAT FN)                                                   (* lmm "10-AUG-78 15:47")
    (PROG ((SKIPEDLEN 0)
           TEM1 TEM2 TAIL)
          (RETURN (COND
                     ((NILPAT (CDR PAT))
                      (MKAND (OR (NOT CHECKINGLENGTH)
                                  (QEQLENGTH VAR (CDAR PAT)))
                             (DOSIDE FN VAR)))
                     [(AND (NULL FN)
                           (COND
                              ([NULLPAT? (SETQ TAIL (SOME (CDR PAT)
                                                          (FUNCTION (LAMBDA (ELT)
                                                                      (COND
                                                                         ((EQ ELT '&)
                                                                          (SETQ SKIPEDLEN (QPLUS 1 SKIPEDLEN))
                                                                          NIL)
                                                                         ((EQ (CAR ELT)
                                                                              '$=)
                                                                          (SETQ SKIPEDLEN (QPLUS SKIPEDLEN
                                                                                                 (CDR ELT)))
                                                                          NIL)
                                                                         (T]
                               [SETQ TEM2 (OR (NOT CHECKINGLENGTH)
                                              (QNOTLESSPLENGTH VAR (QPLUS (CDAR PAT)
                                                                          SKIPEDLEN]
                               (COND
                                  ((CAR INASOME)
                                   (DOWATCH (CAR (FRPLACA INASOME TEM2)))
                                   T)
                                  (T TEM2)))
                              ((NULL TAIL)
                               (QEQLENGTH VAR (QPLUS (CDAR PAT)
                                                       SKIPEDLEN]
                     [(ZEROP (CDAR PAT))
                      (MKAND (DOSIDE FN (QLDIFF VAR VAR))
                             (QMATCHWM VAR (CDR PAT]
                     (T [SETQ TEM1 (COND
                                      ((AND (NUMBERP (CDAR PAT))
                                            (ILESSP (CDAR PAT)
                                                   MAXCDDDDRS))
                                       (QNTH VAR (CDAR PAT)))
                                      (T (SUBSTVAR (QNTH VAR (CDAR PAT]
                        (MKAND (OR (NOT CHECKINGLENGTH)
                                    (NOT (CANMATCHNILLIST (CDR PAT)))
                                    TEM1)
                               (MKAND (DOSIDE FN (QLDIFF VAR (QCDR TEM1)))
                                      (QMATCHWM (QCDR TEM1)
                                             (CDR PAT])

(QMATCHELT1
  [LAMBDA (VAR PAT FN)                                           (* lmm "10-AUG-78 15:47")
    (MKAND [OR (NOT CHECKINGLENGTH)
                (COND
                   ((CDR PAT)
                    (COND
                       ((AND (CANMATCHNIL (CAR PAT))
                             (CANMATCHNILLIST (CDR PAT)))
                        VAR)
                       (T T)))
                   ((CANMATCHNIL (CAR PAT))
                    (QEQLENGTH VAR 1))
                   (T (QNULL (QCDR VAR]
           (MKAND (QMATCHELT (QCAR VAR)
                          (CAR PAT))
                  (MKAND (DOSIDE FN (QCAR VAR))
                         (OR (NULL (CDR PAT))
                             (COND
                                ([AND (EQ (CAR INASOME)
                                          'FASTINASOME)
                                      (COND
                                         [(LISTP (CAR PAT))
                                          (FMEMB (CAAR PAT)
                                                 '(= == %' *ANY* @ SUBPAT]
                                         (T (NOT (FMEMB (CAR PAT)
                                                        '($1 &]
                                 [FRPLACA INASOME (PROG ((INASOME (CONS NIL INASOME)))
                                                        (RETURN (QMATCHWM (QCDR VAR)
                                                                       (CDR PAT]
                                 T)
                                (T (QMATCHWM (QCDR VAR)
                                          (CDR PAT])

(QMATCHELT
  [LAMBDA (VAR PATELT)                                           (* lmm "10-AUG-78 15:47")
                                                                 (* This function matches VAR against PATELT when PATELT is
    a pattern element)
    (COND
       ((NLISTP PATELT)
        (SELECTQ PATELT
            (($ -- &)
                 T)
            (QEQUAL VAR PATELT)))
```

```
         (T (SELECTQ (CAR PATELT)
                  (== (QEQ VAR (CDR PATELT)))
                  (@ [COND
                          [(SIMPLEFN (CADR PATELT))
                           (MKAND (QAPPLY* (CADR PATELT)
                                                  VAR)
                                    (QMATCHELT VAR (CDDR PATELT]
                          (T (MKAND (QMATCHELT VAR (CDDR PATELT))
                                       (QAPPLY* (CADR PATELT)
                                                   VAR])
                  (*ANY* [QOR (MAPCAR (CDR PATELT)
                                         (FUNCTION (LAMBDA (X)
                                                        (QMATCHELT VAR X])
                  (~ (QNOT (QMATCHELT VAR (CDR PATELT))))
                  (%' (QEQUAL VAR (KWOTE (CDR PATELT))))
                  (= (QEQUAL VAR (CDR PATELT)))
                  (SUBPAT (QMATCHSUBPAT VAR (CDR PATELT)))
                  ($= (COND
                           [CHECKINGLENGTH (COND
                                                   (CHECKLENGTH (QEQLENGTH VAR (CDR PATELT)))
                                                   (T (QNOTLESSPLENGTH VAR (CDR PATELT]
                           (T T)))
                  (PATHELP "MATCHELT invalid pattern"])
```

```
(SIMPLEFN
  [LAMBDA (FN)                                                    (* lmm%: "17-NOV-76 19:20:38")

          (* Cheap test if FN is "simple" ; here, just means LISTP NLISTP, EXPRP, LITATOM, etc;
          want to know if it is cheaper to match pattern first, or to check FN first)

     (FMEMB FN SIMPLE.PREDICATES])
```

```
(DOSIDE
  [LAMBDA (WHATTODO X)                                            (* lmm "22-NOV-82 12:24")
     (OR (NULL WHATTODO)
          (MKAND (SELECTQ (CAAR WHATTODO)
                          (<- [OR (CHECKSETQ X WHATTODO)
                                    (MKPROGN (CONS (LIST 'SETQ (CADAR WHATTODO)
                                                              X)
                                                   (AND (CANMATCHNIL (CDDAR WHATTODO))
                                                         (LIST T])
                          (_ (OR (CHECKSETQ X WHATTODO)
                                   (PROGN (DOWATCH (CADAR WHATTODO))
                                           (DOWATCH X)
                                           (PUSH POSTPONEDSETQS (LIST 'SETQ (CADAR WHATTODO)
                                                                             X))
                                           (SETQ LASTEFFECTCANBENIL (CANMATCHNIL (CDDAR WHATTODO)))
                                          T)))
                          (-> (QREPLACE X (CADAR WHATTODO)))
                          (%Ü (DOWATCH (CADAR WHATTODO))
                              (DOWATCH X)
                              (SETQ POSTPONEDRPLACS (CONS (QREPLACE X (CADAR WHATTODO))
                                                                   POSTPONEDRPLACS))
                              T)
                          (@ (QAPPLY* (CADAR WHATTODO)
                                      X))
                          (*GLITCH (FRPLACA (CADAR WHATTODO)
                                              X)
                                     (DOWATCH X)
                                     T)
                          (PATHELP "MATCH FUNARG MISMATCH" WHATTODO))
                   (DOSIDE (CDR WHATTODO)
                           X])
```

```
(CHECKSETQ
  [LAMBDA (X ARGS)
     (COND
         ((FMEMB (CADAR ARGS)
                   %#LIST)
          [COND
              ((FMEMB (CADAR ARGS)
                        %#LISTUSED)
               (MAP INASOME (FUNCTION (LAMBDA (SL)
                                                (AND (OR (EQ (CAR SL)
                                                              'INASOME)
                                                          (EQ (CAR SL)
                                                              'FASTINASOME))
                                                     (RPLACA SL NIL]
          (MAKESUBST (CADAR ARGS)
                   X
                   'WATCH)
          T)
         ((EQ (CADAR ARGS)
               '*)
```

```
        (DOWATCH X)
        (SETQ MUSTRETURN X)
        T])

(DOREPLACE
  [LAMBDA (EXPRESSION SUBSTDONE)
    (PROG NIL
      LP  [SETQ EXPRESSION (OR (DOREPLACE1 (CADR EXPRESSION)
                                      (CADDR EXPRESSION)
                                      (EQ (CAR EXPRESSION)
                                          'TOPREPLACE)
                                      SUBSTDONE)
                               (PROGN [AND (NOT SUBSTDONE)
                                           (SETQ SUBSTDONE T)
                                           (SETQ EXPRESSION (CONS (CAR EXPRESSION)
                                                                  (OR (DOSUBST1 (CDR EXPRESSION))
                                                                      (CDR EXPRESSION]
                                      (GO LP]
          (RETURN (COND
                     (SUBSTDONE EXPRESSION)
                     (T (OR (DOSUBST1 EXPRESSION)
                            EXPRESSION])

(DOREPLACE1
  [LAMBDA (EXPR1 EXPR2 TOPFLG SUBSTDONE)                             (* lmm "10-AUG-78 18:32")
    (OR (EQUAL EXPR1 EXPR2)
        (AND TOPFLG (SELECTQ (CAR EXPR2)
                        ((CONS LIST)
                             (MKAND2 (DOREPLACE1 (QCAR EXPR1)
                                             (CADR EXPR2)
                                             T T)
                                    (OR (AND (EQ (CAR EXPR2)
                                                 'LIST)
                                             (NULL (CDDR EXPR2)))
                                        (DOREPLACE1 (QCDR EXPR1)
                                               (COND
                                                  ((EQ (CAR EXPR2)
                                                       'LIST)
                                                   (CONS 'LIST (CDDR EXPR2)))
                                                  (T (CADDR EXPR2)))
                                               T T))))
                        NIL))
        (SELECTQ (CAR EXPR1)
            (CAR (LIST (LOOK 'RPLACA)
                       (CADR EXPR1)
                       EXPR2))
            (CDR (LIST (LOOK 'RPLACD)
                       (CADR EXPR1)
                       EXPR2))
            (LDIFF (DOREPLACE1 (CADR EXPR1)
                          (QNCONC EXPR2 (CADDR EXPR1))
                          TOPFLG SUBSTDONE))
            (AND SUBSTDONE (LOOKLIST 'RPLNODE2 EXPR1 EXPR2])

)

(DEFINEQ

(PATLEN
  [LAMBDA (PATELT !ED)
    (PROG NIL
      LP  (RETURN (COND
                     [(NLISTP PATELT)
                      (SELECTQ PATELT
                          (($ --)
                               NIL)
                          (& (AND (NOT !ED)
                                       1))
                          (COND
                             (!ED 0)
                             (T 1]
                     (T (SELECTQ (CAR PATELT)
                            (SUBPAT (COND
                                       [!ED (for PE1 in (CDR PATELT) bind (PLEN _ 0) finally (RETURN PLEN)
                                               do (SETQ PLEN (QPLUS PLEN (OR (PATLEN PE1)
                                                                               (RETURN NIL]
                                       (T 1)))
                            ($= (CDR PATELT))
                            ((_ -> <- %Ü @ *GLITCH)
                                 (SETQ PATELT (CDDR PATELT))
                                 (GO LP))
                            (! (SETQ PATELT (CDR PATELT))
                               (SETQ !ED T)
                               (GO LP))
                            (*ANY* (COND
                                      (!ED NIL)
                                      (T 1)))
                            (%’ (COND
                                   (!ED (LENGTH (CDR PATELT)))
                                   (T 1)))
                            ((= == ~)                            (* Currently, ~ can only refer to subpatterns, =, ==, and %’)
                                 (AND (NOT !ED)
                                      1))
                            (($> $<)
                                 NIL)
                            (PATHELP "PATLEN invalid pattern" PATELT])

($?
  [LAMBDA (PATELT)
    (OR (EQ PATELT ’--)
        (EQ PATELT ’$])

(ELT?
  [LAMBDA (PATELT)
    (COND
       [(NLISTP PATELT)
        (OR (NUMBERP PATELT)
            (STRINGP PATELT)
            (FMEMB PATELT ’(& NIL T]
       (T (SELECTQ (CAR PATELT)
               ((= == %’ SUBPAT ~ *ANY*)                          (* Currently, ~ can only refer to =, ==, %’ %, and subpatterns)
                    T)
               ((_ -> <- %Ü @ *GLITCH)
                    (ELT? (CDDR PATELT)))
               NIL])

(SIMPLELT?
  [LAMBDA (PATELT)
    (OR (NLISTP PATELT)
        (SELECTQ (CAR PATELT)
            (@ (SIMPLELT? (CDDR PATELT)))
            ((_ -> <- %Ü)
                NIL)
            T])

(ARB?
  [LAMBDA (PATELT @OKFLG)
    (COND
       ((NLISTP PATELT)
        ($? PATELT))
       (T (SELECTQ (CAR PATELT)
               (! NIL)
               (@ @OKFLG)
               ((<- %Ü _ -> *GLITCH)
                    (ARB? (CDDR PATELT)
                          @OKFLG))
               NIL])

(NULLPAT?
  [LAMBDA (PAT)
    (COND
       ((NULL PAT)
        (NOT CHECKLENGTH))
       (T (EVERY PAT (FUNCTION $?])

(NILPAT
  [LAMBDA (PATLIST)
    (AND CHECKLENGTH (NULL PATLIST])

(CANMATCHNIL
  [LAMBDA (PATELT)

          (* Returns T if PATELT matches NIL, NIL if it doesn’t, and something ELSE
          (maybe) if it might (e.g., =FOO))

    (COND
       ((NLISTP PATELT)
        (AND (FMEMB PATELT ’(& NIL $ --))
            T))
       ((NLISTP (CAR PATELT))
        (SELECTQ (CAR PATELT)
            (@ (AND (CANMATCHNIL (CDDR PATELT))
                    (NOT (FMEMB (CADR PATELT)
                                PATNONNILFUNCTIONS))
                    ’(MAYBE, MAYBE NOT)))
            (SUBPAT (AND (NOT LISTPCHECK)
                         (CANMATCHNILLIST (CDR PATELT))))
            ($< T)
            ($= (OR (NOT (NUMBERP (CDR PATELT)))
                    (ILESSP (CDR PATELT)
                           1)))
            ($> NIL)
            ((_ -> %Ü <- *GLITCH)
                 (CANMATCHNIL (CDDR PATELT)))
            (! [COND
                  ((EQ (CADR PATELT)
                       'SUBPAT)
                   (CANMATCHNILLIST (CDDR PATELT)))
                  (T (CANMATCHNIL (CDR PATELT])
            (%' (NULL (CDR PATELT)))
            ((= ==)
                [NOT (COND
                         [(LITATOM (CDR PATELT))
                          (OR (EQ (CDR PATELT)
                                  T)
                              (AND (CDR PATELT)
                                   (NOT PATVARSNIL]
                         (T (OR (NLISTP (CDR PATELT))
                                (FMEMB (GETP (CAR (CDR PATELT))
                                             'CLISPCLASS)
                                       '(+ * ^ RPLACA RPLACD / - +-))
                                (FMEMB (CAR (CDR PATELT))
                                       NEVERNILFUNCTIONS])
            (*ANY* (SOME (CDR PATELT)
                         (FUNCTION CANMATCHNIL)))
            (~ (CDR PATELT))
            (PATHELP "CANMATCHNIL invalid pattern" PATELT)))
       (T (PATHELP "CANMATCHNIL invalid pattern"])

(CANMATCHNILLIST
  [LAMBDA (PATLIST)
    (EVERY PATLIST (FUNCTION (LAMBDA (PE)
                               (AND (OR (NOT CHECKINGLENGTH)
                                        (NOT (ELT? PE)))
                                    (CANMATCHNIL PE])

(REPLACEIN
  [LAMBDA (PATELT)
    (AND (LISTP PATELT)
         (SELECTQ (CAR PATELT)
             ((-> %Ü *GLITCH)                                        (* the *GLITCH might or might not be a replace, but can't take
                                                                     any chances)
                  T)
             ((@ _ <-)
                  (REPLACEIN (CDDR PATELT)))
             (! (REPLACEIN (CDR PATELT)))
             (SUBPAT (SOME (CDR PATELT)
                           (FUNCTION REPLACEIN)))
             (($= = == %' $< $> ~ *ANY*)                             (* All of these cannot be pointing at a REPLACE)
                  NIL)
             (PATHELP "Invalid pattern REPLACEIN" PATELT])
)

(DEFINEQ

(EASYTORECOMPUTE
  [LAMBDA (EXPRESSION)                                               (* If the EXPRESSION is some cadddaars of a variable, return
                                                                     that variable (something needs to check for VARS bound IN
                                                                     somes and internal forms for WHEN it can't use it for the *'s
                                                                     value))
    (OR (AND (NLISTP EXPRESSION)
             EXPRESSION)
        (AND [OR (GETP (CAR EXPRESSION)
                       'CROPS)
                 (FMEMB (CAR EXPRESSION)
                        '(CAR CDR]
             (EASYTORECOMPUTE (CADR EXPRESSION])

(GENSYML
  [LAMBDA NIL
    (bind TEM until (NOT (FMEMB (SETQ TEM (OR (CAR (SETQ GENSYMVARLIST (CDR GENSYMVARLIST)))
                                              (GENSYM)))
                                VARS))
       finally (RETURN TEM])

(MAKESUBST
  [LAMBDA (VAR VAL FLG)
```

```
    [COND
       ((NULL VAR)
        (SETQ VAR (GENSYML]
    (COND
       ((EQ FLG ’WATCH)
        (DOWATCH VAR)
        (DOWATCH VAL)))
    (SETQ SUBLIST (CONS (CONS VAR (CONS VAL (SELECTQ FLG
                                                 (T T)
                                                 (WATCH (NEQ (EASYTORECOMPUTE VAL)))
                                                 NIL)))
                        SUBLIST))
    VAR])

(DOSUBST
  [LAMBDA (EXPRESSION)

          (* This function does the post substitution in the EXPRESSION;
          it uses SUBLIST to substitute; an entry in SUBLIST is (VAR NEWVALUE . FOUND) where FOUND is initially NIL;
          when the VAR is found for the first time, the FOUND field is smashed with a pointer to that place of substitution;
          then if it is found again, the old place is smashed with a (SETQ $$I VALUE) and then the newvalue is made $$I, and
          "FOUND" is changed to T -
          thus, if an expression occurs once, it is substituted directly; more than once and
          (SETQ $$I -) is put in the first place and $$I in the rest)

    (OR (COND
           [(NLISTP EXPRESSION)
            (CAR (DOSUBST1 (LIST EXPRESSION]
           (T (DOSUBST1 EXPRESSION)))
        EXPRESSION])

(DOSUBST1
  [LAMBDA (EXPRESSION)                                           (* lmm "22-NOV-82 12:24")
    (PROG (TEM1 TEM2)
          (RETURN (COND
                     ((NLISTP EXPRESSION)
                      NIL)
                     [[AND (NLISTP (CAR EXPRESSION))
                           (SETQ TEM1 (find X in SUBLIST suchthat (COND
                                                                     [(NLISTP X)
                                                                      (COND
                                                                         ((EQ X (CAR EXPRESSION))
                                                                          (RETURN]
                                                                     (T (EQ (CAR X)
                                                                            (CAR EXPRESSION]
                                                                 (* (CAR EXPRESSION) needs to be substituted for)
                      (SETQ EXPRESSION (CONS (CAR EXPRESSION)
                                             (CDR EXPRESSION)))
                      [COND
                         ((LISTP (CDDR TEM1))                    (* We have already substituted for it)
                          (SETQ TEM2 (BINDVAR (GENSYML)))
                          (FRPLACA (CDDR TEM1)
                                 (LIST ’SETQ TEM2 (CADDR TEM1)))
                          (FRPLACA (CDR TEM1)
                                 TEM2)
                          (FRPLACD (CDR TEM1)
                                 T)                              (* Mark it that it’s been found twice)
                          )
                         ((NULL (CDDR TEM1))

          (* Haven’t seen it before -
          if CADR TEM1 is NLISTP this means that CAR TEM1 -> CADR TEM1 directly -
          none of this SETQ jazz; so we put T there; otherwise, we save EXPRESSION so that if TEM1%:1 occurs again we can go
          back and wrap setq around the computation of TEM1%:2)

                          (FRPLACD (CDR TEM1)
                                 (COND
                                    ((NLISTP (CADR TEM1))
                                     T)
                                    (T EXPRESSION]
                      (FRPLACA EXPRESSION (CADR TEM1))           (* Might need to substitutions within substituted EXPRESSION)
                      (COND
                         ((NLISTP (CAR EXPRESSION))
                          (OR (DOSUBST1 EXPRESSION)
                              EXPRESSION))
                         (T (FRPLACA EXPRESSION (OR (DOSUBST1 (CAR EXPRESSION))
                                                    (CAR EXPRESSION)))
                            (FRPLACD EXPRESSION (OR (DOSUBST1 (CDR EXPRESSION))
                                                    (CDR EXPRESSION]
                     (T (SELECTQ (CAR EXPRESSION)
                            (LAMBDA

          (* Don’t want to substitute for lambda variables within the lambda;
          this is so that the same variable can be used for a some tail within the some and outside of it)
```

```
                                    [PROG ((SUBLIST (APPEND (CADR EXPRESSION)
                                                           SUBLIST))
                                           TEM)
                                          (RETURN (COND
                                                     ((SETQ TEM (DOSUBST1 (CDDR EXPRESSION)))
                                                      (CONS (CAR EXPRESSION)
                                                            (CONS (CADR EXPRESSION)
                                                                  TEM])
                            (PROG [PROG (V TEM FLG)
                                        [SETQ V (MAPCAR (CADR EXPRESSION)
                                                       (FUNCTION (LAMBDA (X)
                                                                   (COND
                                                                      ([AND (LISTP X)
                                                                            (SETQ TEM (DOSUBST1 (CDR X]
                                                                       (SETQ FLG T)
                                                                       (CONS (CAR X)
                                                                             TEM))
                                                                      (T X]
                                        (RETURN (PROG ((SUBLIST (NCONC [MAPCAR (CADR EXPRESSION)
                                                                              (FUNCTION (LAMBDA (X)
                                                                                          (COND
                                                                                             ((LISTP X)
                                                                                              (CAR X))
                                                                                             (T X]
                                                                       SUBLIST)))
                                                      (RETURN (COND
                                                                 ((OR (SETQ TEM (DOSUBST1 (CDDR EXPRESSION)))
                                                                      FLG)
                                                                  (CONS (CAR EXPRESSION)
                                                                        (CONS V (OR TEM (CDDR EXPRESSION])
                            (QUOTE NIL)
                            ((TOPREPLACE REPLACE)
                                 (DOREPLACE EXPRESSION))
                            (COND
                               [(SELECTQ (CAR EXPRESSION)
                                     ((CAR CDR)
                                         (SELECTQ (CAADR EXPRESSION)
                                              ((CAR CDR)
                                                  T)
                                             NIL))
                                    NIL)
                                (SETQ TEM1 (OR (DOSUBST1 (CADR EXPRESSION))
                                               (CADR EXPRESSION)))
                                (COND
                                   ((EQ (CAR EXPRESSION)
                                        'CDR)
                                    (QCDR TEM1))
                                   (T (QCAR TEM1]
                               (T (PROG (A D)
                                        (SETQ A (DOSUBST1 (CAR EXPRESSION)))
                                        (SETQ D (DOSUBST1 (CDR EXPRESSION)))
                                        (COND
                                           ((EQ (CAR EXPRESSION)
                                                'DUMMY)
                                            (AND D (FRPLACD EXPRESSION D))
                                            (RETURN)))
                                        (RETURN (AND (OR A D)
                                                     (CONS (OR A (CAR EXPRESSION))
                                                           (OR D (CDR EXPRESSION])

(SUBSTVAR
  [LAMBDA (X)                                                     (* lmm%: "27-JUN-77 12:23")
    (MAKESUBST (GENSYML)
          X])

(BINDVAR
  [LAMBDA (VAR VAL)                                               (* lmm "22-NOV-82 12:07")
    (PUSH BOUNDVARS VAR)
    (PUSH BOUNDVALS VAL)
    VAR])

(SELFQUOTEABLE
  [LAMBDA (EXPRESSION)
    (OR (NUMBERP EXPRESSION)
        (STRINGP EXPRESSION)
        (NULL EXPRESSION)
        (EQ EXPRESSION T])

(FINDIN0
  [LAMBDA (VAR X)                                                 (* lmm%: "27-JUN-77 12:23")
    (OR (FINDIN1 VAR X)
        (SOME SUBLIST (FUNCTION (LAMBDA (X)
                                  (AND (FINDIN1 (CAR X)
                                             X)
                                       (FINDIN1 VAR (CDR X])

(FINDIN1
  [LAMBDA (AT LST)                                              (* CHEAP EDITFINDP)
    (OR (EQ AT LST)
        (AND (LISTP LST)
             (OR (FINDIN1 AT (CAR LST))
                 (FINDIN1 AT (CDR LST])

(DOWATCH
  [LAMBDA (X)                                                   (* lmm%: "27-JUN-77 12:23")
    (AND WATCHPOSTPONELST (MAP WATCHPOSTPONELST (FUNCTION (LAMBDA (X)
                                                            (AND (NEQ (CAR X)
                                                                      'FOUND)
                                                                 (FINDIN0 (CAR X)
                                                                        X)
                                                                 (FRPLACA X 'FOUND])

(PATNARGS
  [LAMBDA (X)
    (OR (GETP X 'NARGS)
        (NARGS X])
)

(DEFINEQ

(QNLEFT
  [LAMBDA (EXPRESSION N TAIL NOTFASTFLG)                        (* lmm%: 25-FEB-76 2 19)
    (COND
       (TAIL (LIST (LOOK 'NLEFT)
                   EXPRESSION N TAIL))
       ((ZEROP N)                                               (* NO LOOKUP DONE SINCE FLAST DOESN'T MAKE SENSE
       HERE)
        (LIST 'CDR (LIST 'LAST EXPRESSION)))
       [(EQ N 1)
        (COND
           (NOTFASTFLG (LIST 'LAST EXPRESSION))
           (T (QLAST EXPRESSION]
       (T (LIST (LOOK 'NLEFT)
                EXPRESSION N])

(QNOT
  [LAMBDA (X)
    (QNOT1 X 'NOT])

(QNULL
  [LAMBDA (X)
    (QNOT1 X 'NULL])

(QNOT1
  [LAMBDA (X FNNAME)
    (COND
       ((NLISTP X)
        (SELECTQ X
            ((NIL T)
                 (PATERR "NULL check of T or NIL; possibly a bad pattern"))
            (LIST FNNAME X)))
       (T (SELECTQ (CAR X)
              ((NOT NULL)
                   (CADR X))
              (EQ (FRPLACA X 'NEQ))
              (NEQ (FRPLACA X 'EQ))
              (LISTP (FRPLACA X 'NLISTP))
              (NLISTP (FRPLACA X 'LISTP))
              (LIST FNNAME X])

(QNOTLESSPLENGTH
  [LAMBDA (X N)
    (COND
       ((ZEROP N)
        T)
       (T (QNTH X N])

(QNTH
  [LAMBDA (VAR LEN)
    (COND
       ((OR (NOT (SMALLP LEN))
            (ILESSP LEN 1))
        (LIST (COND
                 (CHECKINGLENGTH (LOOK 'NTH))
                 (T 'FNTH))
              VAR LEN))
       ((IGREATERP LEN MAXCDDDDRS)
        (while (EQ (CAR (LISTP VAR))
                  'CDR)
           do (SETQ LEN (IPLUS LEN 1))
              (SETQ VAR (CADR VAR)))
        (LIST 'NTH VAR LEN))
       (T (while (IGREATERP (SETQ LEN (SUB1 LEN))
                       0)
             do (SETQ VAR (LIST 'CDR VAR)))
          VAR])

(QOR
  [LAMBDA (LISTOFEXPRESSIONS)
    (COND
       ((CDR LISTOFEXPRESSIONS)
        (CONS 'OR LISTOFEXPRESSIONS))
       (T (CAR LISTOFEXPRESSIONS])

(QPLUS
  [LAMBDA (EXPR1 EXPR2)
    (COND
       ((AND (NUMBERP EXPR1)
             (NUMBERP EXPR2))
        (IPLUS EXPR1 EXPR2))
       (T (LIST 'IPLUS EXPR1 EXPR2])

(QREPLACE
  [LAMBDA (VAR EXPRESSION)
    (LIST 'REPLACE VAR EXPRESSION])

(MKAND
  [LAMBDA (X Y)                                                   (* lmm "10-AUG-78 23:00")
    (OR (MKAND2 X Y)
        (LIST 'AND X Y])

(QCAR
  [LAMBDA (X)
    (LIST 'CAR X])

(QCDR
  [LAMBDA (X)
    (LIST 'CDR X])

(QEQ
  [LAMBDA (VAR EXPRESSION)
    (COND
       ((NULL EXPRESSION)
        (QNULL VAR))
       ((ZEROP EXPRESSION)
        (LIST 'ZEROP VAR))
       (T (LIST 'EQ VAR EXPRESSION])

(QEQLENGTH
  [LAMBDA (VAR LEN)                                               (* lmm%: 25-FEB-76 2 10)
    (COND
       ((ZEROP LEN)
        (QNULL VAR))
       ((EQ (CAR (LISTP VAR))
            'CDR)
        (QEQLENGTH (CADR VAR)
               (QPLUS 1 LEN)))
       (T (LIST (LOOK 'EQLENGTH)
                VAR LEN])

(QEQUAL
  [LAMBDA (VAR EXPRESSION)
    [COND
       ([AND (LISTP EXPRESSION)
             (EQ (CAR EXPRESSION)
                 'QUOTE)
             (SELFQUOTEABLE (CAR (LISTP (CDR EXPRESSION]
        (SETQ EXPRESSION (CADR EXPRESSION]
    (COND
       ((NULL EXPRESSION)
        (QNULL VAR))
       ((EQ EXPRESSION T)
        (QEQ VAR EXPRESSION))
       (T (LIST (COND
                   ([OR (SMALLP EXPRESSION)
                        (AND (LISTP EXPRESSION)
                             (EQ (CAR EXPRESSION)
                                 'QUOTE)
                             (LITATOM (CAR (LISTP (CDR EXPRESSION]
                    'EQ)
                   ((NUMBERP EXPRESSION)
                    'EQP)
                   ((STRINGP EXPRESSION)
                    'STREQUAL)
                   (T 'EQUAL))
                VAR EXPRESSION])


(QLAST
  [LAMBDA (X)
    (LIST (LOOK 'LAST X)
          X])


(QAPPLY*
  [LAMBDA (FNNAME VAR)
    (COND
       ((OR (NLISTP FNNAME)
            (EQ (CAR FNNAME)
                'LAMBDA))
        (LIST FNNAME VAR))
       (T (SUBST VAR '@ FNNAME])


(QLDIFF
  [LAMBDA (X Y)                                                 (* lmm%: 25-FEB-76 2 18)
    (LIST (LOOK 'LDIFF)
          X Y])


(QFOR
  [LAMBDA ({OLD} I.V. {ON}VAR {UNTIL}EXPR {FINALLY}EXPR NOSOMEFLG)
                                                                (* lmm "22-NOV-82 12:16")
    (PROG (TEM1)
          (AND (EQ {UNTIL}EXPR T)
               (PATHELP " a SOME with null terminator" (LIST {OLD} I.V. {ON}VAR {FINALLY}EXPR)))
          (AND (EQ {UNTIL}EXPR I.V.)
               (PATERR 'BACKTRACK))
          (AND NOSOMEFLG (GO DOPROG))
          [COND
             ((AND (EQ (CAR (LISTP {UNTIL}EXPR))
                       'AND)
                   (EQ (CADR {UNTIL}EXPR)
                       I.V.))
              (SETQ {UNTIL}EXPR (COND
                                   ((CDDDR {UNTIL}EXPR)
                                    (CONS 'AND (CDDR {UNTIL}EXPR)))
                                   (T (CADDR {UNTIL}EXPR]
          [SETQ TEM1 (OR (SELECTQ (CAR (LISTP {UNTIL}EXPR))
                             (EQ (AND (EQUAL (CADR {UNTIL}EXPR)
                                             (QCAR I.V.))
                                      (LOOKLIST 'MEMB (CADDR {UNTIL}EXPR)
                                             {ON}VAR)))
                             (EQUAL (AND (EQUAL (CADR {UNTIL}EXPR)
                                                (QCAR I.V.))
                                         (LIST 'MEMBER (CADDR {UNTIL}EXPR)
                                               {ON}VAR)))
                             NIL)
                         (LIST 'SOME {ON}VAR
                               (PROG ((ARGS (LIST (GENSYML)
                                                  I.V.)))
                                     (RETURN (LIST 'FUNCTION (COND
                                                                ([AND (EQ (CADR {UNTIL}EXPR)
                                                                          (CAR ARGS))
                                                                      (OR (AND (EQLENGTH {UNTIL}EXPR 2)
                                                                               (EQ (PATNARGS (CAR {UNTIL}EXPR))
                                                                                   1))
                                                                          (AND (EQ (PATNARGS (CAR {UNTIL}EXPR))
                                                                                   1)
                                                                               (EQLENGTH {UNTIL}EXPR 3)
                                                                               (EQ (CADDR {UNTIL}EXPR)
                                                                                   (CADR ARGS]
                                                                 (CAR {UNTIL}EXPR))
                                                                (T (LIST 'LAMBDA ARGS {UNTIL}EXPR]
          (RETURN (COND
                     ((OR {OLD} (NEQ {FINALLY}EXPR T))
                       (MAKESUBST I.V. TEM1)

          (* OLD on means that I.V. is going to be used later on. Thus, we set up to substitute TEM1 for I.V.
          later, and return I.V. now)

                       (RETURN (MKAND I.V. {FINALLY}EXPR)))
                      (T TEM1)))
      DOPROG
          (RETURN `(PROG %, [COND
                                ((NOT {OLD})
                                 (LIST (LIST I.V. {ON}VAR]
                      %,@ [COND
                             ({OLD} `((SETQ %, (BINDVAR I.V.)
                                        %, {ON}VAR]
                      $$SOMELP
                          (COND
                             (%, (NEGATE {UNTIL}EXPR)
                                 (COND
                                    ((LISTP %, I.V.)
                                     (SETQ %, I.V. (CDR %, I.V.))
                                     (GO $$SOMELP)))
                                 (RETURN))
                             (T (RETURN %, {FINALLY}EXPR])

(QLISTP
  [LAMBDA (X)
    (LIST 'LISTP X])

(QNCONC
  [LAMBDA (EXPR1 EXPR2)                                                    (* lmm%: 17 MAY 75 417)
    (COND
       ((NULL EXPR2)
        EXPR1)
       ((EQ (CAR (LISTP EXPR1))
            'LIST)
        (for Y in (REVERSE (CDR EXPR1)) do (SETQ EXPR2 (LIST 'CONS Y EXPR2)))
        EXPR2)
       ((AND (EQ (CAR (LISTP EXPR2))
                 'LIST)
             (NULL (CDDR EXPR2)))
        (LOOKLIST 'NCONC1 EXPR1 (CADR EXPR2)))
       (T (LOOKLIST 'NCONC EXPR1 EXPR2])
)

(DEFINEQ

(PATERR
  [LAMBDA (MSG AT)
    (LISPXPRIN1 (SELECTQ MSG
                     (BACKTRACK "This pattern contains an empty test after a -- or $")
                     (CLISP "The pattern matcher is confused by what it thinks is CLISP
                            within a pattern - please recode this patNIL ")
                     (BADNOT "Cannot negate a non-element pattern")
                     (TWO! "Two !'s in a row")
                     (BAD* "invalid *")
                     (BAD# "invalid #")
                     (BADELT "Pattern item not atom or list ")
                     (NOWITH "no WITH")
                     (AMBIG "ambiguous pattern")
                     (!AT "!atom in middle of pattern")
                     (OR MSG "bad pattern"))
           T)
    (LISPXTERPRI T)
    (COND
       (AT (LISPXPRIN1 " at:    " T)
           (LISPXPRINT AT T T)))
    (LISPXPRIN1 " in:    " T)
    (LISPXPRINT MATCHEXPRESSION T T)
    (ERROR!])

(PATHELP
  [LAMBDA (MESS1 MESS2)
    (LISPXPRIN1 "error in Pattern Match" T)
    (LISPXTERPRI T)
    (HELP MESS1 MESS2])

(LOOKLIST
  [LAMBDA (FN ARG ARG')
    (LIST (LOOK FN ARG ARG')
          ARG ARG'])
```

```
(VALUELOOKUP
  [LAMBDA (VAR)                                                   (* lmm%: 25-FEB-76 2 2)
    (COND
       (LOCALDECLARATION (CLISPLOOKUP0 VAR (CADR MATCHEXPRESSION)
                                NIL LOCALDECLARATION NIL 'VALUE))
       (T (GETATOMVAL VAR])

(LOOK
  [LAMBDA (FN ARG ARG')
    (PROG (CLASS CLASSDEF (LISPFN (OR (GETP FN 'LISPFN)
                                      FN)))
          (RETURN (COND
                     ([AND LOCALDECLARATION (SETQ CLASSDEF (GETP FN 'CLISPCLASSDEF]
                      (CLISPLOOKUP0 FN ARG ARG' LOCALDECLARATION LISPFN (GETP FN 'CLISPCLASS)
                             CLASSDEF))
                     (T LISPFN])
)

(DEFINEQ

(MKAND2
  [LAMBDA (EXPR1 EXPR2)                                           (* lmm "10-AUG-78 23:00")

          (* If the two expressions when ANDed, can be simplified, return the simplified expression otherwise NIL)

    (PROG (TEM TEM2)
          (RETURN (COND
                     ((EQ EXPR1 T)
                      EXPR2)
                     ((EQ EXPR2 T)
                      EXPR1)
                     ((EQUALUNCROP EXPR1 EXPR2)
                      EXPR2)
                     ((EQUALUNCROP EXPR2 EXPR1)
                      EXPR1)
                     (T (OR (SELECTQ (CAR (LISTP EXPR1))
                                 (LISTP (CHECKSLISTP EXPR1 EXPR2))
                                 (PROGN                          (* (AND (AND |...| X) Y) combine X and Y)
                                        (COND
                                           ((SETQ TEM2 (MKAND2 [CAR (SETQ TEM (LAST (LISTP EXPR1]
                                                              EXPR2))
                                            (NCONC1 (LDIFF (LISTP EXPR1)
                                                           TEM)
                                                  TEM2))))
                                 (AND                            (* (AND (AND |...| X) Y) combine X and Y)
                                      [COND
                                         ((SETQ TEM2 (MKAND2 [CAR (SETQ TEM (LAST (LISTP EXPR1]
                                                            EXPR2))
                                          (MKAND [COND
                                                     ((EQ (CDDR (LISTP EXPR1))
                                                          TEM)
                                                      (CADR EXPR1))
                                                     (T (CONS 'AND (LDIFF (CDR (LISTP EXPR1))
                                                                          TEM]
                                                TEM2))
                                         (T (APPEND EXPR1 (LIST EXPR2])
                                 (SETQ (AND (EQUALUNCROP (CADR EXPR1)
                                                  EXPR2)
                                            (SUBST EXPR1 (CADR EXPR1)
                                                  EXPR2)))
                                NIL)
                            (SELECTQ (CAR (LISTP EXPR2))
                                 (AND [COND
                                         [(SETQ TEM (MKAND2 EXPR1 (CADR EXPR2)))
                                          (MKAND TEM (COND
                                                         ((CDDDR EXPR2)
                                                          (CONS 'AND (CDDR EXPR2)))
                                                         (T (CADDR EXPR2]
                                         (T (CONS 'AND (CONS EXPR1 (CDR EXPR2])
                                NIL])

(CHECKSLISTP
  [LAMBDA (EXPR1 EXPR2)                                           (* lmm "10-AUG-78 18:47")

          (* EXPR1 is an expression (LISTP form) - 
          if (AND EXPR1 EXPR2) can be reduced, return the reduced form which returns the same value)

    (COND
       ((EQUAL (CADR EXPR1)
               EXPR2)                                             (* (AND (LISTP X) X) => (LISTP X))
        EXPR1)
       ((NLISTP EXPR2)                                            (* (AND (LISTP X) Y))
        NIL)
       ((SELECTQ (CAR EXPR2)
             ((MEMB MEMBER ASSOC SASSOC)
                 (AND (EQUAL (CADDR EXPR2)
                             (CADR EXPR1))
                      EXPR2))
             ((SOME NLEFT LAST NTH EQLENGTH)
                 (AND (EQUAL (CADR EXPR2)
                             (CADR EXPR1))
                      EXPR2))
            NIL))
       (T (SELECTQ (CAR EXPR2)
               ((CAR CDR FNTH FLAST LISTP NLEFT LAST SOME NTH EQLENGTH)
                   [AND (SETQ EXPR1 (CHECKSLISTP EXPR1 (CADR EXPR2)))
                        (CONS (CAR EXPR2)
                              (CONS EXPR1 (CDDR EXPR2])
               ((EQUAL EQ STREQUAL EQP)
                   [AND (CADDR EXPR2)
                        [OR (SELFQUOTEABLE (CADDR EXPR2))
                            (AND (EQ (CAR (LISTP (CADDR EXPR2)))
                                     'QUOTE)
                                 (CADR (CADDR EXPR2]
                        (SETQ EXPR1 (CHECKSLISTP EXPR1 (CADR EXPR2)))
                        (CONS (CAR EXPR2)
                              (CONS EXPR1 (CDDR EXPR2])
               ((FMEMB FASSOC MEMB MEMBER ASSOC SASSOC)
                   (COND
                      ((SETQ EXPR1 (CHECKSLISTP EXPR1 (CADDR EXPR2)))
                       (LIST (CAR EXPR2)
                             (CADR EXPR2)
                             EXPR1))))
              NIL])

(EQUALUNCROP
  [LAMBDA (EXPR1 EXPR2)                                          (* lmm "10-AUG-78 23:10")
                                                                 (* predicate (AND EXPR1 EXPR2) = EXPR2 -
                                                                 i.e. EXPR2 non-NIL implies EXPR1 non-NIL)

    (OR (EQUAL EXPR1 EXPR2)
        (AND (LISTP EXPR2)
             (COND
                ((GETP (CAR EXPR2)
                       'CROPS)
                 (EQUALUNCROP EXPR1 (CADR EXPR2)))
                (T (SELECTQ (CAR EXPR2)
                        ((CAR CDR NTH NLEFT LAST FLAST FNTH SOME LISTP)
                            (EQUALUNCROP EXPR1 (CADR EXPR2)))
                        ((MEMB FMEMB MEMBER ASSOC SASSOC FASSOC)
                            (EQUALUNCROP EXPR1 (CADDR EXPR2)))
                        ((EQ EQUAL EQP IEQP)
                            (AND [OR (EQ (CADDR EXPR2)
                                         T)
                                     (NUMBERP (CADDR EXPR2))
                                     (AND (EQ (CAR (LISTP (CADDR EXPR2)))
                                              'QUOTE)
                                          (CADR (CADDR EXPR2]
                                 (EQUALUNCROP EXPR1 (CADR EXPR2))))
                       NIL])
)

(DEFINEQ

(PATPARSE
  [LAMBDA (PAT)
    (OR (LISTP PAT)
        (PATHELP "bad input" PAT))
    (PROG (DEFAULTLST)
          (RETURN (PATPARSE1 PAT])

(PATPARSE1
  [LAMBDA (PAT PREFIX)                                           (* DECLARATIONS%: UNDOABLE)
                                                                 (* lmm%: "27-JUN-77 12:35")
    (PROG (TEM TEM2 TEM3 CARPAT CDRPAT NOTFOUND)
          (OR PAT (RETURN))
      RETRY
          [AND (CDR PAT)
               (NLISTP (CDR PAT))
               (SETQ PAT (LIST (CAR PAT)
                               '%.
                               (CDR PAT]                         (* Take care of (a . b) by changing it to
                                                                 (a %. b))
          (SETQ CARPAT (CAR PAT))
          [AND (EQ CARPAT COMMENTFLG)
               (NULL NORMALCOMMENTSFLG)
               (SETQ CARPAT (CAR (GETCOMMENT PAT]
          (SETQ CDRPAT (CDR PAT))
          [COND
       [(LISTP CARPAT)
        (SELECTQ (CAR CARPAT)
             (*ANY* [SETQ CARPAT (CONS (CAR CARPAT)
                                       (PROG ((TOPPAT CARPAT))
                                             (RETURN (PATPARSE1 (CDR CARPAT]
                    (OR (EVERY CARPAT (FUNCTION SIMPLELT?))
                        (PATERR "*ANY*/*EVERY* construct too compicated" PAT)))
             (QUOTE

    (* This is so (-- (QUOTE A) --) means (-- %' A --); this kludge is necessary now since DWIMIFY1B sometimes parses the %'
    A into (QUOTE A))

                    [COND
                       ((NOT (ATOM (CADR CARPAT)))
                        (/RPLNODE PAT '%' (CONS (CADR CARPAT)
                                                CDRPAT)))
                       (T (/RPLACA PAT (PACK (LIST '%' (CADR CARPAT]
                    (GO RETRY))
             (LAMBDA                                       (* (-- (LAMBDA (X) --) --) means (--
                                                           &@ (LAMBDA (X) --)))
                     (/ATTACH '&@ PAT)
                     (GO RETRY))
             (PROGN                                        (* Otherwise, it's a sub-pattern)
                    (SETQ CARPAT (MAKESUBPAT (PROG ((TOPPAT CARPAT))
                                                    (RETURN (PATPARSE1 CARPAT]
       ((NOT (LITATOM CARPAT))                             (* Strings and numbers parse to themselves)
        (OR (STRINGP CARPAT)
            (NUMBERP CARPAT)
            (PATERR 'BADELT CARPAT)))
       (T (SELECTQ CARPAT
               ((T NIL & -- $))
               ($$ (SETQQ CARPAT --))
               ($1 (SETQQ CARPAT &))
               (($2 $3 $4 $5 $6 $7 $8 $9)
                    (SETQ CARPAT (CONS '$= (NTHCHAR CARPAT 2))))
               ((== = $> $< $=)
                   (SETQ TEM2 (PATGETEXPR CDRPAT PAT))
                   [SETQ CARPAT (COND
                                    ((AND (EQ CARPAT '$=)
                                          (EQ (CAR TEM2)
                                              1))
                                     '&)
                                    (T (CONS CARPAT (CAR TEM2]
                   (SETQ CDRPAT (CDR TEM2)))
               ((! %.)
                   (SETQ TEM2 (PATPARSE1 CDRPAT))
                   (RETURN (CONS (MAKE!PAT (CAR TEM2)
                                          TEM2 PAT PREFIX)
                                 (CDR TEM2))))
               (~ (SETQ TEM2 (PATPARSE1 CDRPAT))
                  (RETURN (CONS (NEGATEPAT (CAR TEM2)
                                         PAT)
                                (CDR TEM2))))
               (%' (SETQ CARPAT (CONS '%' (CAR CDRPAT)))
                   (SETQ CDRPAT (CDR CDRPAT)))
               (COND
                  ((SETQ TEM (PATUNPACK PAT))
                   (SETQ PAT TEM)                          (* Now, either we have a "DEFAULT" condition, or else a var
                                                           infix condition)
                   (GO RETRY))
                  (T (SETQ NOTFOUND PAT]

    (* By now, CARPAT is set to the parsing of the first thing in PAT;
    and CDRPAT is the appropriate tail; want to check for infix operators;
    if NOTFOUND is non-nil, then CARPAT was an atom which wasn't parseable as a pattern;
    might be a variable if followed by a _ or a %# or a *)

REINFIX
    [COND
       ((AND CDRPAT (NLISTP CDRPAT))
        (SETQ CDRPAT (LIST '%. CDRPAT]
    (COND
       ((SETQ TEM (AND CDRPAT (FASSOC (CAR CDRPAT)
                                      PATTERNREPLACEOPRS)))
        [COND
           [NOTFOUND

    (* CARPAT is not a pattern, and followed by a _; want to know if the next thing is a pattern or something else;
    it is assumed that var_pattern is meant; I could change it to mean pat_var)

                  [COND
                     ((FMEMB CARPAT %#LIST))
                     ((STRPOS "#" CARPAT 1 NIL 1)
                      (SETQ %#LIST (CONS CARPAT %#LIST]    (* Check if a %# type variable)
                  (SETQ TEM3 (PATPARSE1 (CDR CDRPAT)
                                        CDRPAT))
                  (RETURN (CONS (CONS (CADR TEM)
                                         (CONS CARPAT (CAR TEM3)))
                                   (CDR TEM3]
           (T (SETQ CARPAT (CONS (CADDR TEM)
                                 (CONS [CAR (SETQ CDRPAT (PATGETEXPR (CDR CDRPAT]
                                       CARPAT)))
              (SETQ CDRPAT (CDR CDRPAT]
        (GO REINFIX))
       (NOTFOUND (COND
                    ((AND (EQ (NTHCHAR (CAR CDRPAT)
                                      1)
                               '_)
                           (IGREATERP (NCHARS (CAR CDRPAT))
                                 1))
                     (/RPLNODE CDRPAT '_ (CONS (MKATOM (SUBSTRING (CAR CDRPAT)
                                                              2 -1))
                                               (CDR CDRPAT)))
                     (GO REINFIX)))
              (COND
                 (PREFIX (PATERR (COND
                                      ((STRPOSL CLISPCHARRAY (CAR PAT))
                                         'CLISP)
                                      (T 'AMBIG))
                                PAT)))
              (SETQ PAT (PARSEDEFAULT PAT NIL PREFIX))
              (SETQ NOTFOUND)
              (GO RETRY))
       ((EQ (CAR CDRPAT)
            '@)
        (SETQ CDRPAT (OR (PATUNPACKINFIX1 (CDR CDRPAT))
                         (CDR CDRPAT)))
        (SETQ CARPAT (CONS '@ (CONS (PATGETFNNAME CDRPAT)
                                        CARPAT)))
        (SETQ CDRPAT (CDR CDRPAT))
        (GO REINFIX))
       ((SETQ TEM (PATUNPACKINFIX CDRPAT))
        (SETQ CDRPAT TEM)
        (GO REINFIX)))
    (RETURN (CONS CARPAT (PATPARSE1 CDRPAT])
```

```
(PATUNPACKINFIX1
  [LAMBDA (L)
    (PATPARSEAT L PATTERNINFIXES1])
```

```
(PARSEDEFAULT
  [LAMBDA (PAT LOCALVARDEFAULT PREFIX)                            (* lmm "22-MAY-80 21:37")

          (* Turns PAT%:1 (which is a LITATOM) into the "DEFAULT" pattern -
          I.e. PAT%:1 couldn't be parsed as a pattern -
          It is assumed that the default for an atom is an element pattern)

    (OR (AND (LITATOM (CAR PAT))
             (NEQ (CAR PAT)
                  T)
             (CAR PAT))
        (PATHELP "MAKEDEFAULT" (CAR PAT)))
    (PROG (SMASHFLG NEWPAT)
          (COND
             ((FMEMB (CAR PAT)
                     DEFAULTLST)                                  (* Second occurance of a "DEFAULT" is defaulted to =)
              (SETQQ LOCALVARDEFAULT =))
             ([COND
                 ((STRPOS "#" (CAR PAT)
                         1 NIL 1)
                  (OR (NUMBERP (SUBATOM (CAR PAT)
                                      2 -1))
                      (PATERR 'BAD# PAT)))
                 ((STRPOS "*" (CAR PAT))
                  (OR (EQ (CAR PAT)
                          '*)
                      (PATERR 'BAD* PAT]                          (* %#n is defaulted to _ the first time)
              (SETQQ LOCALVARDEFAULT SETQ))
             ((AND (NLISTP (CAR PAT))
                   (STRPOSL CLISPCHARRAY (CAR PAT)))
              (PATERR 'CLISP PAT)))
      RETRY
          [SETQ NEWPAT (SELECTQ (OR LOCALVARDEFAULT (AND (NLISTP VARDEFAULT)
                                                         VARDEFAULT))
                            ((_ SETQ SET)
                                (SETQ DEFAULTLST (CONS (CAR PAT)
                                                       DEFAULTLST))
                                [CONS (CAR PAT)
                                      (CONS '_ (CONS '& (CDR PAT])
                            ('%'
                                [COND
                                   (SMASHFLG (/ATTACH '%' PAT))
                                   (T (RETURN (CONS '%' PAT])
                            ((= EQUAL)
                                [COND
                                   (SMASHFLG (/ATTACH '= PAT))
                                   (T (RETURN (CONS '= PAT])
                            ((== EQ)
                                [COND
                                   (SMASHFLG (/ATTACH '== PAT))
                                   (T (RETURN (CONS '== PAT])
                            ((@ APPLY*)
                                [COND
                                   (SMASHFLG (/ATTACH '$1@ PAT))
                                   (T (RETURN (CONS '$1 (CONS '@ PAT])
                            (PROGN (SETQ SMASHFLG T)
                                   [SETQ LOCALVARDEFAULT (COND
                                                            (LOCALVARDEFAULT (PATERR (COND
                                                                                         (VARDEFAULT "invalid
                                                                                                PATTERNVARDEFAULT
")
                                                                                         (T 'AMBIG))
                                                                                    PAT))
                                                            ((EQ 1 (GETP (CAR PAT)
                                                                         'NARGS))
                                                             (SETQ SMASHFLG)
                                                             '@)
                                                            ((VARCHECK (CAR PAT)
                                                                    T T T)
                                                             '=)
                                                            ((LISTP VARDEFAULT)
                                                             (CAR VARDEFAULT))
                                                            (T '?]
                                   (GO RETRY]
          (COND
             (SMASHFLG (/RPLNODE2 PAT NEWPAT)
                    (RETURN PAT))
             (T (RETURN NEWPAT])

(VARCHECK
  [LAMBDA (VAR NOMESSFLG SPELLFLG PROPFLG)

          (* Checks if VAR is really a variable -
          Used by MAKEDEFAULT to avoid bad parsings)

    (OR (AND (LITATOM VAR)
             (OR (FMEMB VAR VARS)
                 (NEQ (EVALV VAR)
                      'NOBIND))
             VAR)
        (AND (NOT NOMESSFLG)
             (ERROR VAR "NOT A VARIABLE" T])

(PATUNPACK
  [LAMBDA (PAT)                                                    (* lmm "22-MAY-80 21:37")

          (* THIS WOULD BE SIMPLER IF THERE WERNT THINGS LIKE $N AROUND --
          THIS FUNCTION UNPACKS (CAR PAT) ALONG THE LINES OF PATTERN OPERATORS -
          I'LL MAKE IT SIMPLER BY ASSUMING THAT THINGS ARE OK
          (I.E. WILL UNPACK) (AND (STRPOSL PATTERNCHARRAY (CAR PAT))
          (PROG ((CHARS (UNPACK (CAR PAT))) RESULTS) RETRY (for CHR on CHARS do
          (for X in PATCHRLST bind TAIL do (SETQ TAIL CHR) (COND
          ((for Z in (CDR X) always (COND ((EQ Z (CAR TAIL)) (SETQ TAIL
          (CDR TAIL)) T))) (* CHARS IS (|...| PATCHRSTRING |...|); WE TAKE AND PUT ON RESULTS THE UNPACKING OF THE
          FIRST AND REST) (SETQ RESULTS (NCONC RESULTS (COND
          ((NEQ CHR CHARS) (LIST (PACK (LDIFF CHARS CHR)))) (T NIL))
          (LIST (CAR X)))) (SETQ CHARS TAIL) (GO RETRY))))) (RETURN
          (AND RESULTS (NCONC1 RESULTS (PACK CHARS)) (RETURN RESULTS))))))

    (PATPARSEAT PAT PATCHARS])

(PATUNPACKINFIX
  [LAMBDA (L)
    (PATPARSEAT L PATTERNINFIXES1])

(PATGETFNNAME
  [LAMBDA (L)                                                      (* wt%: "14-JUN-78 10:59")
    (OR (LISTP (CAR L))
        (FGETD (CAR L))
        (FIXSPELL (CAR L)
               70 SPELLINGS2 T L (FUNCTION GETD)
               NIL NIL T)
        (FIXSPELL (CAR L)
               70 USERWORDS T L (FUNCTION GETD)
               T))
    (CAR L])

(PATGETEXPR
  [LAMBDA (L UP)                                                (* lmm%: "19-SEP-76 23:26:14")
    (OR L (PATERR "missing an expression" UP))
    (SETQ L (OR (PATUNPACKINFIX L)
                L))
    [COND
       ((LISTP (CAR L))
        (PROG ((VARS (APPEND %#LIST VARS)))
              (RETURN (DWIMIFY0? (CAR L)
                             (CAR L)
                             NIL NIL NIL FAULTFN]
    (for X in %#LIST when (AND (NOT (FMEMB X %#LISTUSED))
                              (FINDIN1 X (CAR L)))
       do (SETQ %#LISTUSED (CONS X %#LISTUSED)))
    L])

(PATPARSEAT
  [LAMBDA (PAT CHRS)                                            (* lmm "22-MAY-80 21:38")

          (* Breaks apart (CAR PAT) if possible, replaces the parsing into the beginning of PAT ;
          otherwise return NIL if can't -
          CHRS is a list of args as if to STRPOS, i.e. check (STRPOS X%:1 PAT%:1 1 NIL X%:2) for X in CHRS -
          X%:1 is the char list, X%:2 is ANCHOR)

    (PROG (TEM DONEANYTHING LST POS)
          (OR (AND (NLISTP (CAR PAT))
                   (STRPOSL PATTERNCHARRAY (CAR PAT)))
              (RETURN))
          (SETQ LST (UNPACK (CAR PAT)))
      LP  (COND
             ((NULL CHRS)
              (RETURN))
             ((EQ (CADDR (CAR CHRS))
                  (CAR PAT))
              (RETURN))
             ([NOT (SETQ POS (COND
                                [(NULL (CADAR CHRS))
                                 (find X on LST suchthat (for Z in (CAAR CHRS) as ZZ in X
                                                            always (EQ Z ZZ]
                                ((for Z in (CAAR CHRS) as ZZ in LST always (EQ Z ZZ))
                                 LST]
              (SETQ CHRS (CDR CHRS))
              (GO LP)))

          (* Found one -
          POS is now the tail of LST which begins with one of the operators)

          [SETQ PAT (CONS (CAR PAT)
                          (COND
                             ([SETQ TEM (FNTH POS (ADD1 (FLENGTH (CAAR CHRS]
                              (CONS (PACK TEM)
                                    (CDR PAT)))
                             (T (CDR PAT]
          [SETQ TEM (COND
                       ([AND TEM (EQ (CADDR (CAR CHRS))
                                     '$)
                             (NOT (FMEMB (CAR TEM)
                                         '(_ @ = < >]
                        '$=)
                       (T (CADDR (CAR CHRS]
          (COND
             [(NEQ POS LST)
              (RPLNODE PAT (PACKLDIFF LST POS)
                   (CONS TEM (CDR PAT]
             (T (FRPLACA PAT TEM)))
          (RETURN PAT])

(MAKE!PAT
  [LAMBDA (PATELT PATALL REALPAT PREFIX)
    (COND
       ((AND (EQ (CAR REALPAT)
                 '!)
             (EQ PATELT (CAR PATALL))
             (OR (EQ (CAR PATELT)
                     '_)
                 (EQ (CAR PATELT)
                     '<-))
             (NOT (FMEMB (CADR PATELT)
                         DEFAULTLST)))                          (* Change PATALL to ((_ var ! subpat %.
                                                                all of it)) from ((_ var . part1) part2))
        [FRPLACD (CDR PATELT)
               (MAKE!PAT (MAKESUBPAT (CONS (CDDR PATELT)
                                             (CDR PATALL]
        (FRPLACD PATALL NIL)
        PATELT)
       (T (OR (COND
                 ((NLISTP PATELT)
                  (SELECTQ PATELT
                      (& '$)
                      (($ --)
                          '$)
                      NIL))
                 (T (SELECTQ (CAR PATELT)
                        (! (PATERR 'TWO! PATELT))
                        ((_ <- %Ü -> @)
                             (FRPLACD (CDR PATELT)
                                    (MAKE!PAT (CDDR PATELT)))
                             PATELT)
                        (* (CONS (CAR PATELT)
                                 (MAKE!PAT (CDR PATELT))))
                        (SUBPAT (AND (NULL (CDDR PATELT))
                                     (NOT (ELT? (CADR PATELT)))
                                     (CADR PATELT)))
                        ($= PATELT)
                        NIL)))
              (CONS '! PATELT])

(MAKESUBPAT
  [LAMBDA (PATLST)
    (COND
       ((NULL PATLST)
        NIL)
       ([OR (EQUAL PATLST '(--))
            (EQUAL PATLST '($]
        '&)
       (T (CONS 'SUBPAT PATLST])

(NEGATEPAT
  [LAMBDA (PE REALPAT)
    (PROG NIL
          [COND
             ((NLISTP PE)
              (SELECTQ PE
                  ((& $)
                       (PATERR "Cannot negate this type of pattern" PE))
                  T))
             (T (SELECTQ (CAR PE)
                    ((= == %' SUBPAT))
                    ((_ %Ü <- ->)
                         [RETURN (CONS (CAR PE)
                                       (CONS (CADR PE)
                                             (NEGATEPAT (CDDR PE])
                    (@)
                    (PATERR 'BADNOT REALPAT]
          (RETURN (CONS '~ PE])

(PACKLDIFF
  [LAMBDA (LST1 LST2)
    (PROG (TEM1 TEM2)
          (FRPLACD (OR (SETQ TEM1 (NLEFT LST1 1 LST2))
                       (HELP))
                 NIL)
          (RETURN (PROG1 (PACK LST1)
                         (FRPLACD TEM1 TEM2])
)

(RPAQQ PATCHARS
       ((($ <)
         T $<)
        (($ >)
         T $>)
        (($ =)
         T $=)
        ((%')
         T %')
        ((!)
         T !)
        ((= =)
         T ==)
        ((=)
         T =)
        ((~)
         T ~)
        ((< -)
         NIL <-)
        ((@)
         NIL @)
        ((_)
         NIL _)
        (($)
         T $)))

(RPAQQ PATTERNINFIXES (((_)
                        T _)
                       ((< -)
                        T <-)
                       ((@)
                        T @)))

(RPAQQ PATTERNINFIXES1 (((_)
                         NIL _)
                        ((< -)
                         NIL <-)
                        ((@)
                         NIL @)))

(RPAQQ PATTERNREPLACEOPRS ((_ _ %Ü)
                           (__ <- ->)
                           (_!!_!_ _ %Ü)
                           (<- <- ->)))

(RPAQQ PATTERNITEMS
       ((&)
        (--)
        ($$ --)
        (T)
        (NIL)
        (&)
        (--)
        ($)
        ($1 &)
        ($2 ($= . 2))
        ($3 ($= . 3))
        ($4 ($= . 4))
        ($5 ($= . 5))
        ($6 ($= . 6))))

(RPAQQ NEVERNILFUNCTIONS (CONS LIST QUOTE ABS ADD1 SUB1 CONCAT REMAINDER FREMAINDER IREMAINDER LOGOR LOGAND
                          LOGXOR))

(RPAQQ PATNONNILFUNCTIONS (GETD NUMBERP STRINGP ZEROP LISTP SMALLP))

(RPAQ PATTERNCHARRAY [MAKEBITTABLE (NCONC (MAPCAR PATCHARS 'CAAR)
                                          (MAPCAR PATTERNITEMS 'CAR])

(RPAQQ PATGENSYMVARS (GENSYMVARS%: $$1 $$2 $$3 $$4 $$5 $$6 $$7 $$8 $$9 $$10 $$11 $$12 $$13 $$14 $$15 $$16 $$17
                      ))

(RPAQQ PATVARDEFAULT =)

(RPAQQ MAXCDDDDRS 5)

(RPAQQ PATCHECKLENGTH T)

(RPAQ PATLISTPCHECK (EQ 'VAX (SYSTEMTYPE)))

(RPAQQ PATVARSMIGHTBENIL T)

(RPAQQ PATCHARS
       ((($ <)
         T $<)
        (($ >)
         T $>)
        (($ =)
         T $=)
        ((%')
         T %')
        ((!)
         T !)
        ((= =)
         T ==)
        ((=)
         T =)
        ((~)
         T ~)
        ((< -)
         NIL <-)
        ((@)
         NIL @)
        ((_)
         NIL _)
        (($)
         T $)))

(RPAQQ PATTERNINFIXES (((_)
                        T _)
                       ((< -)
                        T <-)
                       ((@)
                        T @)))

(RPAQQ PATTERNINFIXES1 (((_)
                         NIL _)
                        ((< -)
                         NIL <-)
                        ((@)
                         NIL @)))

(RPAQQ PATTERNREPLACEOPRS ((_ _ %Ü)
                           (__ <- ->)
                           (_!!_!_ _ %Ü)
                           (<- <- ->)))

(RPAQQ PATTERNITEMS
       ((&)
        (--)
        ($$ --)
        (T)
        (NIL)
        (&)
        (--)
        ($)
        ($1 &)
        ($2 ($= . 2))
        ($3 ($= . 3))
        ($4 ($= . 4))
        ($5 ($= . 5))
        ($6 ($= . 6))))

(RPAQQ NEVERNILFUNCTIONS (CONS LIST QUOTE ABS ADD1 SUB1 CONCAT REMAINDER FREMAINDER IREMAINDER LOGOR LOGAND
                               LOGXOR))

(RPAQQ PATNONNILFUNCTIONS (GETD NUMBERP STRINGP ZEROP LISTP SMALLP))

(RPAQQ SIMPLE.PREDICATES (LISTP LITATOM NLISTP CAR CDR NULL))

(RPAQ PATTERNCHARRAY [MAKEBITTABLE (NCONC (MAPCAR PATCHARS 'CAAR)
                                          (MAPCAR PATTERNITEMS 'CAR])

(RPAQQ PATGENSYMVARS (GENSYMVARS%: $$1 $$2 $$3 $$4 $$5 $$6 $$7 $$8 $$9 $$10 $$11 $$12 $$13 $$14 $$15 $$16 $$17
                            ))

(OR (BOUNDP 'MATCHSTATS)
    (SETQ MATCHSTATS))

(RPAQQ PATVARDEFAULT =)

(RPAQQ MAXCDDDDRS 5)

(RPAQQ PATCHECKLENGTH T)

(RPAQQ PATLISTPCHECK NIL)

(RPAQQ PATVARSMIGHTBENIL T)

(RPAQQ MATCHBLOCKS
       ((MATCHBLOCK (ENTRIES MAKEMATCH)
               (GLOBALVARS PATCHARS MAXCDDDDRS PATNONNILFUNCTIONS PATGENSYMVARS PATTERNREPLACEOPRS
                      PATTERNINFIXES1 PATTERNCHARRAY NEVERNILFUNCTIONS MATCHSTATS SIMPLE.PREDICATES USERWORDS
                      SPELLINGS2 CLISPCHARRAY NORMALCOMMENTSFLG COMMENTFLG)
               (LOCALFREEVARS WATCHPOSTPONELST SUBLIST INASOME CHECKINGLENGTH WMLST LASTEFFECTCANBENIL
                      POSTPONEDSETQS MUSTRETURN BOUNDVARS BOUNDVALS GENSYMVARLIST SKIPEDLEN ZLENFLG
                      LOCALDECLARATION MATCHEXPRESSION MATCHEFFECTS CHECKLENGTH %#LIST %#LISTUSED PATVARSNIL
                      POSTPONEDRPLACS LISTPCHECK DEFAULTLST VARDEFAULT)
               (SPECVARS EXPR FAULTFN VARS CLISPCHANGE)
              MAKEMATCH QMATCHSUBPAT QMATCHWM QMATCH$ QMATCH! QMATCH$= QMATCHELT1 QMATCHELT SIMPLEFN DOSIDE
              CHECKSETQ DOREPLACE DOREPLACE1 PATLEN $? ELT? SIMPLELT? ARB? NULLPAT? NILPAT CANMATCHNIL
              CANMATCHNILLIST REPLACEIN EASYTORECOMPUTE GENSYML MAKESUBST DOSUBST DOSUBST1 SUBSTVAR BINDVAR
              SELFQUOTEABLE FINDIN0 FINDIN1 DOWATCH PATNARGS QNLEFT QNCONC QNOT QNULL QNOT1 QNOTLESSPLENGTH
              QNTH QOR QPLUS QREPLACE MKAND QCAR QCDR QEQ QEQLENGTH QEQUAL QLAST QAPPLY* QLDIFF QFOR QLISTP
              PATERR PATHELP LOOKLIST VALUELOOKUP LOOK MKAND2 CHECKSLISTP EQUALUNCROP PATPARSE PATPARSE1
              PATUNPACKINFIX1 PARSEDEFAULT VARCHECK PATUNPACK PATUNPACKINFIX PATGETFNNAME PATGETEXPR PATPARSEAT
              MAKE!PAT MAKESUBPAT NEGATEPAT PACKLDIFF)))

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: MATCHBLOCK (ENTRIES MAKEMATCH)
       (GLOBALVARS PATCHARS MAXCDDDDRS PATNONNILFUNCTIONS PATGENSYMVARS PATTERNREPLACEOPRS PATTERNINFIXES1
              PATTERNCHARRAY NEVERNILFUNCTIONS MATCHSTATS SIMPLE.PREDICATES USERWORDS SPELLINGS2 CLISPCHARRAY
              NORMALCOMMENTSFLG COMMENTFLG)
       (LOCALFREEVARS WATCHPOSTPONELST SUBLIST INASOME CHECKINGLENGTH WMLST LASTEFFECTCANBENIL POSTPONEDSETQS
              MUSTRETURN BOUNDVARS BOUNDVALS GENSYMVARLIST SKIPEDLEN ZLENFLG LOCALDECLARATION MATCHEXPRESSION
              MATCHEFFECTS CHECKLENGTH %#LIST %#LISTUSED PATVARSNIL POSTPONEDRPLACS LISTPCHECK DEFAULTLST
              VARDEFAULT)
       (SPECVARS EXPR FAULTFN VARS CLISPCHANGE)
       MAKEMATCH QMATCHSUBPAT QMATCHWM QMATCH$ QMATCH! QMATCH$= QMATCHELT1 QMATCHELT SIMPLEFN DOSIDE CHECKSETQ
       DOREPLACE DOREPLACE1 PATLEN $? ELT? SIMPLELT? ARB? NULLPAT? NILPAT CANMATCHNIL CANMATCHNILLIST REPLACEIN
       EASYTORECOMPUTE GENSYML MAKESUBST DOSUBST DOSUBST1 SUBSTVAR BINDVAR SELFQUOTEABLE FINDIN0 FINDIN1 DOWATCH
       PATNARGS QNLEFT QNCONC QNOT QNULL QNOT1 QNOTLESSPLENGTH QNTH QOR QPLUS QREPLACE MKAND QCAR QCDR QEQ
       QEQLENGTH QEQUAL QLAST QAPPLY* QLDIFF QFOR QLISTP PATERR PATHELP LOOKLIST VALUELOOKUP LOOK MKAND2
       CHECKSLISTP EQUALUNCROP PATPARSE PATPARSE1 PATUNPACKINFIX1 PARSEDEFAULT VARCHECK PATUNPACK PATUNPACKINFIX
       PATGETFNNAME PATGETEXPR PATPARSEAT MAKE!PAT MAKESUBPAT NEGATEPAT PACKLDIFF)
)

(PUTPROPS MATCH COPYRIGHT ("Venue & Xerox Corporation" 1982 1984 1990))
```

{MEDLEY}<library>MATCH.;1 28-Jun-2024 18:34:03

-- Listed on 30-Jun-2024 13:13:12 --

FUNCTION INDEX

VARIABLE INDEX