DATA

LINK L1 TO ANY * I2 TO ANY * I3 TO ANY
TYPE CONS IS
   CAR TO ANY
   CDR TO ANY
STRING LNIL IS "NIL"

END

ROUTINE FLATTEN( X TO ANY ) TO STRING IS NAME( X )
ROUTINE LIST•STR( X TO ANY, FLAG TO BOOLEAN FALSE ) TO STRING IS
   LINK T TO ANY
   IF T=LNIL THEN RETURN \( )\)
   IF TYPE•OF( X )=CONS THEN RETURN FLATTEN( X )
   RESULT = LIST•STR( CAR OF X ) ; T ; CDR OF X
   IF T=LNIL THEN GOTO EXIT
   IF TYPE•OF( T )=CONS THEN RESULT = RESULT ;\( )\) ; FLATTEN( T ) ;\( )\)
   RESULT = RESULT ;\( )\) ; LIST•STR( T, TRUE )
   IF NOT FLAG THEN RESULT = RESULT \( )\) \( )\)
   RETURN

EXIT;

NEXT•TOKEN•LIST( NXT•TOK TO ROUTINE ) TO ANY IS
   NEXT•TOK IS A PARAMETER; THIS ROUTINE EXPECTED TO RETURN ONE OF:
   \( )\) BEGINNING OF LIST
   \( )\) END OF LIST
   \( )\) DELIMITER FOR DOTTED PAIR
   ANYTHING ELSE -- AN ATOM

THE LANGUAGE ACCEPTED BY TOKEN•LIST IS THAT OF <S•EXP> BELOW:

<EXPR> ::= <ATOM> \( ()\) \( ()\)
<EXPR•SEQ> ::= <S•EXPR> \( \) \( \) \( \) \( \)

ROUTINE TOKEN•LIST
   LINK T = NXT•TOK()
   IF T=\( ()\) OR T=\( ()\) THEN GOTO ERROR
   IF T=\( ()\) THEN RETURN T
   RETURN TOKEN•LIST( NXT•TOK )\( )\)

END

ROUTINE TOKEN•LIST0( NXT•TOK TO ROUTINE ) TO ANY IS
   LINK T = NXT•TOK(), L TO CONS
   IF T=\( ()\) THEN RETURN LNIL
   IF T=\( ()\) THEN GOTO ERROR
   RESULT = CREATE( CONS ) \( )\)
   LOOP:
   IF T=\( ()\) THEN
      CAR OF L \( )\) TOKEN•LIST0( NXT•TOK )\( )\)
   ELSE
      CAR OF L = T
   ENDIF
   NXT•TOK()
   IF T=\( ()\) THEN CDR OF L = LNIL \( )\)
   IF T=\( ()\) THEN GOTO LOOP
   CDR OF L = TOKEN•LIST( NXT•TOK )\( )\)
   IF NXT•TOK()=\( ()\) THEN RETURN ELSE GOTO ERROR
ROUTINE CLEARNOUDE (M) TO NODE IS
  LINK S
FOR S IN SFL (M) DO DETACH (S, M) END
DATA IN P11, P1
TYPE NODESET IS
(NODE1 TO INTEGER
NODESET ALREADY
LINK T TO INTEGER + 0
END
ROUTINE P11(T) TO STRING IS
LINK SELS
LINK NODE
IF T FROM ALREADY THEN
IF NAME(T) = NODE THEN
RETURN NODE + NAME(T OF ALREADY)
ELSE
RETURN NAME(T)
ENDIF
ENDIF
IF (T IS NODE) AND (T IS T) THEN
[I+1]
ATTACH(T, ALREADY)
T OF ALREADY + 1
ENDIF
IF T IS ATOM THEN RETURN NAME(T)
IF T IS LINK THEN RETURN + P11(S1)
IF T IS SEQ THEN
IF S IN T DO
IF S = C OF FIRST OP T THEN PAI +
RAD11(S)
ENDIF
RETURN PA
ENDIF
ELSIF SELS =
OR S IN SELS DO
IF S = C OF FIRST OP SELS THEN PAI +
RAD11(S OF T)
ENDIF
RETURN PA
END
ROUTINE P1(T) IS
LEARNODE(ALREADY)
+ 0
ND
DATA IN P12,P2
TYPE NODESET IS
ENDOF1 TO INTERP
NODESET ALREADY
LINK T TO INTERP & T
END
ROUTINE P12(T) TO STRING IS
LINK SFLS
LINK P* S
IF T FROM ALREADY THEN
IF NAME(T)=NODE(T) THEN
RETURN NODE(T) NAME(T OF ALREADY)
ELSE
RETURN NAME(T)
ENDIF
ENDIF
IF (T IS NODE) AND (T=UND) THEN

ATTACH(T ALREADY)
T OF ALREADY = T
ENDIF
IF T IS ATOM THEN RETURN NAME(T)
IF T IS LINK THEN RETURN P* P12($T)
IF T IS SFD THEN
R$.P($)
FOR S+ IN T DO
IF SELC OF FIRST OF T THEN R$R$R$R$($S)
ENDFOR
RETURN R$.P
ENDIF
R$.P($)
SFLS = SFL(T)
FOR S+ IN SFLS DO
IF S+ = C OF FIRST OF SFLS THEN R$R$R$R$($S)
ENDFOR
RETURN R$.P
END
CLEARMODE (ALREADY)
I*O
PRINT($P12(T))
END
ROUTINE PACK(S TO SEP) TO STRING IS
  LINK T
  RESULT → T
  FOR T IN S DO
    IF T IS STRING THEN RESULT = RESULT AT T ELSE ERROR
  ENDFOR
END
ROUTINE SCAN() TO SEGTS
LINK CH TO STRING, WORD TO STRING ""
RESULT = CREATE (SEG)
FOR CH IN UNPACK(READ()) DO
   IF CHER THEN
      IF WORD = "" THEN ADDL(WORD, RESULT); WORD = ""
      ELSE
         WORD = WORD CH
      ENDIF
   ENDIF
ENDFOR
END
ROUTINE ADDF(A,S TO S') TO
LINK T TO LISTEL
T = FIRST OF S
FIRST OF S = CREATE(LISTEL)
C OF FIRST OF S = A
H OF FIRST OF S = T
IF T = UND THEN LAST OF S = FIRST OF S
END
ROUTINE DELF(S TO S'N) TO ANY IS
IF FIRST OF S = END THEN RETURN
RESULT = C OF FIRST OF S
FIRST OF S = NEXT OF FIRST OF S
END
ROUTINE ADDLI (A, S TO SED) IS
IF LAST OF S E THEN
  N OF LAST OF S = CREATE(LISTEL)
  LAST OF S = N OF LAST OF S
  C OF LAST OF S = A
ELSE
  LAST OF S = CREATE(LISTEL)
  FIRST OF S = LAST OF S
  C OF LAST OF S = A
ENDIF
ROUTINE DEL((S TO SEQ)) TO ANY IS
LINK T TO LISTEL.
IF FIRST OF S = UNIF THEN ERROR
FOR T = FIRST OF S, Y OF T WHILE Y OF T = LAST OF S DO NOOP()
RESULT = C OF LAST OF S
LAST OF S = T
END