

Information International, Inc.
23 August 1965

LISP II MEMO #13
PROGRAMMING EXAMPLES

Michael Levin

Abstract: This memo contains examples of LISP II Programs. Each program is commented, and equipped with a driver that tests it on some data. The machine's output is indicated by †. The main functions of each program are then shown translated into internal language.

These programs have not been debugged, and may contain conceptual as well as syntactic errors.

EXAMPLE 1

```
FUNCTION PATH(GRAPH, START, FINISH) FLUID GRAPH, FINISH;
  REVERSE(PATH1(LIST(START)));
```

```
% PATH FINDS A PATH FROM START TO FINISH ALONG THE DIRECTED
% GRAPH CALLED GRAPH. START AND FINISH ARE NODES OF THE
% GRAPH. THE PATH IS A LIST OF NODES. IF THERE IS NO
% PATH POSSIBLE, THEN PATH HAS VALUE NIL.
```

```
FUNCTION PATH1(WAY) IF CAR WAY=FINISH THEN WAY ELSE
  BEGIN SYMBOL X,Y; FOR X IN GRAPH DO IF CAR X=CAR WAY
    AND NOT MEMBER(CDR X,WAY) AND Y=PATH1(CDR X . WAY)
  RETURN Y END;
```

```
FUNCTION REVERSE(L) REVI(L,NIL);
```

```
FUNCTION REVI(R,L) IF NULL L THEN R ELSE REVI(CDR L,CAR L . R);
```

```
FUNCTION MEMBER(X,L) L AND (X=CAR L OR MEMBER(X, CDR L));
```

```
SYMBOL G1;
```

```
G1=((A . C) (C . B) (B . A) (C . D) (D . E))
†((A . C) (C . B) (B . A) (C . D) (D . E))
```

```
PATH(G1,'B','E')
†(B A C D E)
```

```
PATH(G1,'D','C')
†()
```

```
STOP
```

```
(FUNCTION PATH ((GRAPH FLUID) START (FINISH FLUID))
  (REVERSE (PATH1 (LIST START))))
```

```
(FUNCTION PATH1 (WAY) (IF (EQUAL (CAR WAY) FINISH) WAY
  (BLOCK ((X SYMBOL) (Y SYMBOL)) (FOR X IN GRAPH
    (IF (AND (EQUAL (CAR X) (CAR WAY)) (NOT (MEMBER
      (CDR X) WAY)) (SET Y (PATH1 (CONS (CDR X) WAY))))
    (RETURN Y))))))
```

```
(FUNCTION REVERSE (L) (REVI L NIL))
```

```
(FUNCTION REVI (L R) (IF (NULL L) R (REVI (CDR L) (CONS (CAR L) R))))
```

```
(FUNCTION MEMBER (X L) (AND L (OR (EQUAL X (CAR L)) (MEMBER X (CDR L))))))
```

EXAMPLE 2

```
NOVALUE FUNCTION MATRIXMULTIPLY(X,Y,Z,L,M,N) REAL ARRAY X,Y,Z;
  INTEGER L,M,N; LOC Z; BEGIN INTEGER I,J,K;
% MATRIXMULTIPLY MULTIPLIES TWO GIVEN MATRICIES X AND Y. IT
%   CREATES A NEW ARRAY TO HOLD THE VALUE, AND PLACES THIS ARRAY
%   IN Z.
Z=MAKEARRAY(L,N,'REAL);
FOR I=1 STEP 1 UNTIL L DO
  FOR K=1 STEP 1 UNTIL N DO
    FOR J=1 STEP 1 UNTIL M DO
      Z(I,K)=Z(I,K) + X(I,J)*Y(J,K)  END;
REAL ARRAY A=[REAL [2. 0.] [0. 2.]],
  B=[REAL [0. 1.] [1. 0.]], C;
MATRIXMULTIPLY(A,B,C,2,2,2);
C;
†[REAL [0. 2.] [2. 0.]]
STOP
```

```

(FUNCTION (MATRIXMULTIPLY NOVALUE) ((X (ARRAY REAL)) (Y (ARRAY REAL))
  (Z (ARRAY REAL) LOC) (L INTEGER) (M INTEGER) (N INTEGER))
(BLOCK ((I INTEGER) (J INTEGER) (K INTEGER))
(SET Z (MAKEARRAY L N (QUOTE REAL)))
(FOR I 1 STEP 1 UNTIL L
  (FOR K 1 STEP 1 UNTIL N
    (FOR J 1 STEP 1 UNTIL M
      (SET (Z I K) (PLUS (Z I K) (TIMES (X I J) (Y J K))))))))))

```

EXAMPLE 3

```

INTEGER FUNCTION SUMSQUARE(X) INDEF X(I); BEGIN INTEGER Y,J;
  FOR J=1 STEP 1 UNTIL I DO Y=Y+X(J);
  RETURN Y END;

```

```

SUMSQUARE (3,4,5);

```

```

$ 50

```

```

STOP

```

```

(FUNCTION (SUMSQUARE INTEGER) ((X INDEF I))
  (BLOCK ((Y INTEGER) (J INTEGER))
    (FOR J 1 STEP 1 UNTIL I (SET Y (PLUS Y (X J))))
    (RETURN Y) ))

```

EXAMPLE 4

```

FUNCTION MAPLIST (X, FN) FLUID X; FORMAL(SYMBOL, SYMBOL) FN;
  IF NULL X THEN NIL ELSE
    FN(X) . MAPLIST (CDR X, FN);

```

```

FUNCTION JX(L, X) MAPLIST (L, FUNCTION(()) (K) CAR K . X, X));

```

% IN THIS EXAMPLE, X IN MAPLIST IS FLUID WITHOUT GOOD REASON.
 % INSPITE OF THIS, THE CORRECT FREE X IN THE FUNCTIONAL
 % ARGUMENT IS OBTAINED.

```

JX(' (A B C), 'M)
† ((A . M) (B . M) (C . M))

```

STOP

```

(FUNCTION MAPLIST ((X FLUID) (FN (FORMAL SYMBOL SYMBOL)))
  (IF (NULL X) NIL (CONS (FN X) (MAPLIST (CDR X) FN))))

```

```

(FUNCTION JX (L X) (MAPLIST L (FUNCTION () (K)
  (CONS (CAR J) X) X)))

```