QUANTUM THEORY PROJECT
FOR RESEARCH IN ATOMIC, MOLECULAR, AND SOLID STATE
CHEMISTRY AND PHYSICS
UNIVERSITY OF FLORIDA, GAINESVILLE, FLORIDA

OUTPUT FUNCTIONS FOR MBLISP

By

Harold V. McIntosh

PROGRAM NOTE   # 17

5 September 1963

ABSTRACT

By using the data movement and output operators in MBLISP it
is possible to devise specialized output functions, capable of pro-
ducing a variety of output formats. Two such functions are described.
The first, COMPACTIFY, removes all redundant blanks from LISP source
language, and therefore may be used to compress a static and tested
program for economy of loading. The second, IPRINT, is designed to
produce output in a more legible format by indenting subexpressions
and spacing lines according to certain key words.

ACKNOWLEDGEMENTS

Gainesville, 5 September 1963

IPRINT

IPRINT is available as a compactified LISP deck, between two
eared cards of which the program to be IPRINTed is inserted. IPRINT*,
defined in terms of (LAMBDA L ...) is the function to be applied in this
case.

Indentation of two columns will occur whenever any of the key
words specified by the function (KEYWORD X) occur. At present these are
AND and OR, but the list may be changed at will.

Indentation together with a skipped line will occur whenever any
of the key words specified by the function (SPACEKEY X) occur. These
are presently DEFINE, COND, and RPPLY.

The amount of indentation is controlled by the number of
INDENTMORE and INDENTLESS cards which appear in the definition of IPACKEXPR.
One could also modify the definition of these functions themselves.

Column 0 is automatically filled with a blank so as to cause the
program-controlled printer to operate correctly.

IPRINT has been used to prepare the listing of IPRINT and
COMPACTIFY.

FUNCTIONS DEFINED ...   SET VAL SEQ XEC WRIBU ///W PACKATOM PACKATOM
* NBLANKS INDENT INDENTMORE INDENTLESS WRITELINE IPRINT IPRINT* IPACKEX
PR SPACEKEY KEYWORD IPACKSTRING IPACKLIST IPACKSPACEDSTRING

```
(DEFINE

(SET (LAMBDA (X Y) (SAR X (CIDR Y))))

(VAL (LAMBDA* (X) (CAIDR X)))

(SEQ (LAMBDA* (X) (XAR ((CADR X) (CAIDR X)) (CIDR X))))

(XEC (LAMBDA* (X Y) (SAR (X (CAIDR Y)) (CIDR Y))))

(WRIBU (LAMBDA (X) (IF (EQ X ($WRIBU1)) ($WRIBU2) ($WRIBU1))))

(///W (LAMBDA () (IF (PACK (BLANK)) (///W) (AND
    ($WRITE (SEQ WRIBU))
    (PACSET (VAL WRIBU))
    (INDENT)))))

(PACKATOM (LAMBDA (X) (AND
    (DISSET X)
    (PACKATOM* (DISINT)))))

(PACKATOM* (LAMBDA (X) (OR
    (NULL X)
    (AND
      (OR
        (PACK X)
        (WRITELINE))
      (PACKATOM* (DISINT))))))

(NBLANKS (LAMBDA (N) (OR
    (ZER N)
    (AND
      (PACK (BLANK))
      (NBLANKS (DECR N))))))

(INDENT (LAMBDA () (NBLANKS (VAL INDENT))))

(INDENTMORE (LAMBDA () (XEC INCR INDENT)))

(INDENTLESS (LAMBDA () (XEC DECR INDENT)))

(WRITELINE (LAMBDA () (AND
    ($WRITE (SEQ WRIBU))
    (PACSET (VAL WRIBU))
    (INDENT))))

(IPRINT (LAMBDA (E) (AND
    (SET ($WRIBU1) (QUOTE WRIBU))
    (PACSET (VAL WRIBU))
    (SET (DEC (QUOTE 1)) (QUOTE INDENT))
    (INDENT)
    (IPACKEXPR E)
```

```
   (///W))))

(IPRINT* (LAMBDA L (AND
   (SET ($WRIBU1) (QUOTE WRIBU))
   (PACSET (VAL WRIBU))
   (SET (DEC (QUOTE 1)) (QUOTE INDENT))
   (INDENT)
   (IPACKSPACEDSTRING L)
   (///W))))

(IPACKEXPR (LAMBDA (E) (OR
   (AND
     (ATOM E)
     (PACKATOM E))
   (AND
     (PACKATOM (LPAREN))
     (OR
       (NULL E)
       (AND
           (KEYWORD (CAR E))
           (PACKATOM (CAR E))
           (INDENTMORE)
           (INDENTMORE)
           (IPACKSTRING (CDR E))
           (INDENTLESS)
           (INDENTLESS))
         (AND
           (SPACEKEY (CAR E))
           (PACKATOM (CAR E))
           (INDENTMORE)
           (INDENTMORE)
           (IPACKSPACEDSTRING (CDR E))
           (INDENTLESS)
           (INDENTLESS))
         (AND
           (IPACKEXPR (CAR E))
           (IPACKLIST (CDR E))))
       (PACKATOM (RPAREN))))))

(SPACEKEY (LAMBDA (X) (OR
   (EQ X (QUOTE DEFINE))
   (EQ X (QUOTE COND))
   (EQ X (QUOTE APPLY)))))

(KEYWORD (LAMBDA (X) (OR
   (EQ X (QUOTE AND))
   (EQ X (QUOTE OR)))))

(IPACKSTRING (LAMBDA (S) (OR
   (NULL S)
   (AND
     (///W)
     (IPACKEXPR (CAR S))
     (IPACKSTRING (CDR S))))))

(IPACKLIST (LAMBDA (L) (OR
   (NULL L)
   (AND
     (PACKATOM (BLANK))
```

```
    (IPACKEXPR (CAR L))
    (IPACKLIST (CDR L))))))

(IPACKSPACEDSTRING (LAMBDA (S) (OR
  (AND
    (NULL S)
    (///W)
    (///W))
  (AND
    (///W)
    (///W)
    (IPACKEXPR (CAR S))
    (IPACKSPACEDSTRING (CDR S))))))

)
```

COMPACTIFY

COMPACTIFY is available as a compactified LISP deck, between two eared cards of which the program to be compactified is placed. Any string of expressions may be compactified; the string will be treated as a list, between elements of which blank cards will be placed. Should a compactified deck end in colimn 72, an additional blank card will result. The output appears on PCHTAP (B-4) and uses all 72 columns.

All redundant blanks---namely all those except those which separate two consecutive atoms---are removed from the unput deck. Consequently the function may be used to process well-established LSSP functions for economy of loading, storage, or mailing.

UNCTIONS DEFINED ...   SET VAL SEQ XEC WRIBU ///W PACKATOM PACKATOM
NBLANKS INDENT INDENTMORE INDENTLESS WRITELINE IPRINT IPRINT* IPACKEX
R SPACEKEY KEYWORD IPACKSTRING IPACKLIST IPACKSPACEDSTRING

DEFINE

(SET (LAMBDA (X Y) (SAR X (INCR (CDR Y)))))

(VAL (LAMBDA* (L) (CAR (INCR (CDR L)))))

(SEQ (LAMBDA* (L) (XAR ((CADR L) (CAR (INCR (CDR L)))) (INCR (CDR L))
)))

)

DEFINE

(PCHBU (LAMBDA (X) (IF (EQ X ($WRIBU1)) ($WRIBU2) ($WRIBU1))))

(/// (LAMBDA () (AND
   (PACK (BLANK))
   (PACK (BLANK))
   (///))))

(COMPACTIFY (LAMBDA L (AND
   (SET ($PCHBU1) (QUOTE PCHBU))
   (PACSET (VAL PCHBU))
   (NOT (///))
   ($PUNCH (SEQ PCHBU))
   (PACSET (VAL PCHBU))
   (COMPACTIFY* L)
   (NOT (///))
   ($PUNCH (SEQ PCHBU))
   (PACSET (VAL PCHBU)))))

(COMPACTIFY* (LAMBDA (L) (OR
   (NULL L)
   (AND
     (NOT (///))
     ($PUNCH (SEQ PCHBU))
     (PACSET (VAL PCHBU))
     (COMPEXPR (CAR L))
     (NOT (///))
     ($PUNCH (SEQ PCHBU))
     (PACSET (VAL PCHBU))
     (COMPACTIFY* (CDR L))))))

(PUNCHATOM (LAMBDA (X) (AND
   (DISSET X)
   (PUNCHATOM* (DISINT)))))

(PUNCHATOM* (LAMBDA (X) (OR
   (NULL X)
   (AND
     (OR
       (PACK X)

```
        (AND
            ($PUNCH (SEQ PCHBU))
            (PACSET (VAL PCHBU))))
        (PUNCHATOM* (DISINT)))))))

(COMPLIST (LAMBDA (L) (OR
    (NULL L)
    (AND
        (ATOM (CAR L))
        (PUNCHATOM (CAR L))
        (COMPLIST* (CDR L)))
    (AND
        (COMPEXPR (CAR L))
        (COMPLIST (CDR L))))))

(COMPLIST* (LAMBDA (L) (OR
    (NULL L)
    (AND
        (ATOM (CAR L))
        (PUNCHATOM (BLANK))
        (PUNCHATOM (CAR L))
        (COMPLIST* (CDR L)))
    (AND
        (COMPEXPR (CAR L))
        (COMPLIST (CDR L))))))

(COMPEXPR (LAMBDA (E) (OR
    (AND
        (ATOM E)
        (PUNCHATOM E))
    (AND
        (PUNCHATOM (LPAREN))
        (COMPLIST E)
        (PUNCHATOM (RPAREN)))))))

)
```