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 producing 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

The verification of these two functions was done on the IBM 709 computer of the University of Florida computer center. I am grateful to the committee supervising the operation of the computer for the provision of the requisite time.

Harold V. McIntosh

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 APPLY.

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
R SPACEKEY KEYWORD IPACKSTRING IPACKLIST IPACKSPACEDSTRING

DEFINE

{SET (LAMBDA (X Y) (SAR X (CIDR Y)))}
{VAL (LAMBDA* (X) (CAIDR X))}
{SEQ (LAMBDA* (X) (XAR ((CAIDR X) (CAIDR X)) (CAIDR X)))}
{XEC (LAMBDA* (X Y) (SAR (X (CAIDR Y)) (CAIDR 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
TOR
{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)
\((/ / W))\)

\((I P R I N T* (L A M B D A \( L \) (A N D
\( (S E T (\$ W R I B U 1) (Q U O T E W R I B U))\)
\( (P A C S E T (V A L W R I B U))\)
\( (S E T (D E C (Q U O T E 1)) (Q U O T E I N D E N T))\)
\( (I N D E N T)\)
\( (I P A C K S P A C E D S T R I N G \( L \))\)
\((/ / W))\))\)

\((I P A C K E X P R (L A M B D A (E)) (O R
\( (A N D
\( (A T O M E))\)
\( (P A C K A T O M (L P A R E N))\)
\( (O R
\( (N U L L E))\)
\( (A N D
\( (K E Y W O R D (C A R E))\)
\( (P A C K A T O M (C A R E))\)
\( (I N D E N T M O R E)\)
\( (I N D E N T M O R E)\)
\( (I P A C K S P A C E D S T R I N G (C D R E))\)
\( (I N D E N T L E S S)\)
\( (I N D E N T L E S S)\)
\( (A N D
\( (S P A C E K E Y (C A R E))\)
\( (P A C K A T O M (C A R E))\)
\( (I N D E N T M O R E)\)
\( (I N D E N T M O R E)\)
\( (I P A C K S P A C E D S T R I N G (C D R E))\)
\( (I N D E N T L E S S)\)
\( (I N D E N T L E S S)\)
\( (A N D
\( (I P A C K E X P R (C A R E))\)
\( (I P A C K L I S T (C D R E)))\)
\( (P A C K A T O M (R P A R E N)))))\)

\((S P A C E K E Y (L A M B D A \( X \)) (O R
\( (E Q X (Q U O T E D E F I N E))\)
\( (E Q X (Q U O T E C O N D))\)
\( (E Q X (Q U O T E A P P L Y)))\))\)

\((K E Y W O R D (L A M B D A \( X \)) (O R
\( (E Q X (Q U O T E A N D))\)
\( (E Q X (Q U O T E O R)))\))\)

\((I P A C K S T R I N G (L A M B D A \( S \)) (O R
\( (N U L L S))\)
\( (A N D
\( (/ / W)\)
\( (I P A C K E X P R (C A R S))\)
\( (I P A C K S T R I N G (C D R S))))))\)

\((I P A C K L I S T (L A M B D A \( L \)) (O R
\( (N U L L L))\)
\( (A N D
\( (P A C K A T O M (B L A N K))))\)
(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 column 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 LISP functions for economy of loading, storage, or mailing.
DEFINE

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

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

(seq (LAMBDA* (L) (XAR ((CAR (CDR L) (CAR (INCR (CDR L))))) (INCR (CDR L))))

DEFINE

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

(/(/ (LAMBDA () (AND

car (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)))))

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

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

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

1
