

LISP II PROJECT

MEMORANDUM 7

I. Boolean Connectives

Evaluation is from left to right only as far as necessary.
Any nm-Boolean argument is equivalent to TRUE.

B1 AND B2... (AND B1 B2...)
AND, OR, XOR Have an indefinite number of args.
NOT B (NOT B)
B1 IMPLIES B2 (IMPLIES B1 B2)
IMPLIES , EQUIV have 2 arguments.

II. Bit Operators

The values are octal. The operation is dependent on word length.

(LOGAND A1 A2...) performs parallel anding of bits LOGAND.
LOGOR, LOGXOR have an indefinite number of arguments.

(INVERT A) inverts all bits
(SHIFT A I) shifts A left for positive I.
(CYCLE A I) cycles A left for positive I.

III. Relations

Both arguments are arithmetic. The value is Boolean.

EQUAL = (works for all types of arguments)
LEQ <=
LESS <
GEQ >=
GREATER >
NEQ / (works for all types of arguments)
EQUALN (works for all types of arguments, but numbers must be of same type for it to be TRUE)

IV. Arithmetic

Types may be mixed freely.

(PLUS A1 A2...)	$A1 + A2 + \dots$
(TIMES A1 A2...)	$A1 * A2 * \dots$
(MINUS A)	$- A$
(DIFFERENCE A1 A2)	$A1 - A2$
(QUOTIENT A1 A2)	$A1/A2$ (value is real)
(REMAINDER I1 I2)	$I1 I2$ (value is integer)
(IQUOTIENT I1 I2)	$I1 : -I2$ (value is integer)
(SIN R)	(value is real)
(COS R)	(value is real)
(LOG R) (log base c)	(value is real)
(EXP R) e^R	
(EXPT A1 A2)	$A1 \uparrow A2$
(FLOAT I)	
(ROUND R)	
(ENTER R)	
(SCALE R I)	$R * 2^I$ (performs arithmetic shifting)

V. Lisp

$C \langle A | D \rangle * R$ (may be used as locatives)

CONS, LIST, APPEND

ATOM, EQ, NULL

TRUTH NOT NULL (X)

NILCDR NULL (CDR(X))

LENGTH (of list)

VI. Atom Classifiers

IDP

 NUMBERP
 REALP
 INTEGERP
 OCTALP
 BOOLEANP
 

(These are true for arrays of given type.)

ARRAYP
STRINGP**VII. Arrays**

ARYDIM (M) How many dimensions? Value is integer.

ARYSIZE (M,I) Size of ith dimension of M. Value is integer.

 ARRAY (T, D1,...DN) creates a blank array
 T=Type REAL, SYMBOL, ETC.
 D1 - size of each dimension

 VECTOR (S1,...SN) creates a 1-dimensional
 SYMBOL, ARRAY with elements S1.
VIII. Strings

STRINGL (T) How many characters? Value is integer.

EXPAND (T) Value is a list of character identifiers.

COMPRESS (S) Compresses a list of character identifiers into a string.

FINDID (T) A predicate, Is there an identifier with same spelling.

GETID (T) Value is identifier with same spelling.

GETST (S) Argument is an identifier. Value is string with same spelling.

CONCAT (T1,...TN) Concatenates n strings into 1 string.

INPRINT (S) If S is any atom other than an array the value is a string which is its print name.

SYMPRINT(S) Same as inprint, but can always be read in.

INPUT#A'##
%#))#
3.4INPRINTA#
)
3.4SYMPRINT#A¹##
%#))#
3.4

Other special number printers will be available,
 e.g., REALFORMAT (X,3,9) prints the BCD for X using
 9 characters, 3 of which are to the left of the decimal point.

IV. Property Lists

PROP(S) If S is an atom, then PROP(S) is its property list. Property lists are initially null. PROP(S) may be used as a locative.

X. System cheaters

Use with caution!

OCTREP(S)	Converts a character identifier to an octal number.
CHID.(Q)	Does the opposite.
WORD.(S)	Gets the word pointed to by S. Value is octal.
CONSW.(Q)	Puts the quantity Q in list space and returns a pointer.
PQC.(S)	Does nothing. The argument is of type symbol, the value is the same pointer as an octal number.
STORE.(Q,S)	Stores Q in the location pointed to by S.
QPC.(S)	is the inverse of QPC.

XI. Conversion

1. Full words to lists.

ISC.	Converts an integer into a listed integer.
RSC.	Same for reals.
QSC.	Same for octals.

2. Listed numbers into full words.

SRC.	Converts a listed quantity to a real if possible. Otherwise it gives diagnostics.
SIC.	Converts a listed quantity to an integer if possible. Otherwise, it gives diagnostics.