

September 16, 1968

SMILE AT LISP

Lynn Quam

ABSTRACT:

This memo is intended to explain the use of the LISP functions in the system file SYS: SMILE.LSP. These functions are a generally useful extension to the functions already defined in the LISP interpreter, and are discussed in six categories.

- I. Functions for defining functions
- II. Functions for debugging functions
- III. Functions for printing the definitions of functions
- IV. Functions for using output devices
- V. Functions for disk input
- VI. Other useful functions

To get these functions into the LISP interpreter, evaluate
(INC(INPUT SYS: SMILE))

I. FUNCTIONS FOR DEFINING FUNCTIONS

DE DF and DM are FEXPRs used to define EXPRs FEXPRs and MACROS. They have the side effect of ENTERING the name of the function into a list called ALLFNS. DE DF and DM also tell you when you have REDEFINED a function by returning the list (<name> REDEFINED) instead of <name>. The calling sequence for DE is:

(DE <name> <variable list> <function body>)

Example:

To define an EXPR whose name is SIZE:

you type:

```
(DE SIZE (L) (COND ((ATOM L)  $\emptyset$ ) (T (PLUS 1 (SIZE (CAR L)) (SIZE (CDR L))))))
```

lisp types:

SIZE

the definition of SIZE is then:

```
(DEFPROP SIZE  
(LAMBDA (L)  
(COND ((ATOM L)  $\emptyset$ ) (T (PLUS 1 (SIZE (CAR L)) (SIZE (CDR L))))))  
EXPR)
```

DF and DM behave similarly.

SAILON No. 41

DV is a FEXPR useful for defining a VALUE. DV has the side effect of entering the name of the VALUE defined into a list called ALLVALUES.

The calling sequence for DV is:
(DV <name> <unquoted value>)

Example:

you type:
(DV X (VALUE OF X))

lisp types:
(VALUE OF X)

the definition of X is then:
(DEFPROP X
(NIL VALUE OF X)
VALUE)

See appendix I for definitions of DE DF DM and DV.

II. FUNCTIONS FOR DEBUGGING FUNCTIONS

Three different types of debugging aids are available: TRACE, TRACET and BREAK.

A) TRACE and its auxillary functions UNTRACE and RESET allow one to monitor the entrance to and exit from "traced" functions. (Warning: use (NOUO T) with compiled functions) when a "traced" function is entered,

(ENTERING <recursion depth> <function name>) <values of arguments>
is typed. When exited,

(LEAVING <recursion depth> <function name>) <result>
is typed. Appendix II contains an example of trace applied to the function SIZE.

(TRACE <list of names>) FEXPR causes all functions in list of names to be "traced." TRACE returns a list of names of those functions which were previously not traced.

(UNTRACE <list of names>) FEXPR is the inverse to TRACE, ie., it restores each function to its previous untraced state.

(RESET) EXPR causes all recursion depth counters to be reset to zero. Only necessary when a traced function is abnormally exited.

B) TRACET and its auxillary functions UNTRACET SLST and USLST allow one to monitor all SET's or SETQ's to atoms selected for by SLST. When such a SET or SETQ occurs,

(SET <atom name> <value>) or
(SETQ <atom name> <value>)

SAILON No. 41

is printed. (Warning - use (NOUUD T) with compiled functions)
(TRACET) EXPR turns on SET-SETQ monitoring.
(UNTRACET) EXPR turns off SET-SETQ monitoring.
(SLST <list of atoms>) FEXPR Appends <list of atoms> to the list of
monitored atoms.
(USLST <list of atoms> FEXPR Removes each atom from list of monitored
atoms.

C) (BREAK <break pt. #> <value>) EXPR is useful for observing the
state of variable bindings within lambda expressions and
progs. When BREAK is entered, (BREAK <break pt. #>) is
printed. BREAK then enters a READ-EVAL-PRINT loop until
semicolon is typed to READ. BREAK then exits with <value>.

III. FUNCTIONS FOR PRINTING DEFINITIONS OF FUNCTIONS AND OTHER PROPERTIES.

(GRINDEF <list of atoms?>) FEXPR is used to print properties of atoms
in readable DEFPROP format. GRINDEF produces output in
which parenthesis depth and matching is recognizable by
line indentation.

The normal list of properties which GRINDEF prints
is (EXPR FEXPR VALUE MACRO). Any non-atomic member of
<list-or-atoms?> designates a different list of properties.
Appendix I was produced by:

```
(GRINDEF DE DF DM DV ENTER)
```

(GRINL <atom>) FEXPR causes all atoms in the list (<atom> <value of atom>
to be GRINDEFed.

For example, (GRINL ALLFNS) will cause ALLFNS and every function which
has been defined by DE DF or DM to be GRINDEFed.

GRINDEF uses the auxillary functions SPRINT HUNOZ PANL and PPOS.

IV. FUNCTIONS FOR USING OUTPUT DEVICES

(LPT) EXPR is used to start an output file on the line printer. It does
(PROG NIL (OUTC (OUTPUT LPT:) T)
(LINELENGTH LPTLENGTH)
(OUTTIME))

where OUTTIME prints a heading, time and date.

SAILON No. 41

(OFF) EXPR is used to end an output file. It does
(PROG NIL (PRINT T)
(OUTC NIL T)
(LINELENGTH TTYLENGTH))

(LPTOUT <expr-list>) FEXPR is used to create an entire output file on the lineprinter. It does
(PROG NIL (LPT)
(MAPC (FUNCTION EVAL) <expr-list>)
(OFF))

Examples: (LPTOUT (GRINL ALLFNS))
(LPTOUT (PRINT OBLIST) (PRINT FOO))

(DSKOUT <file name> <expr-list>) FEXPR is used to create an entire output file on disk file DSK: <file-name>.LSP. It sets linelength to LPTLENGTH, and evaluates all expressions in <expr-list>, then does (OFF).

Example: (DSKOUT NEWFNS (GRINL NEWFNS))

Appendix IV contains complete definitions.

V. FUNCTIONS FOR USING INPUT DEVICES

(DSKIN <list of file-names>) FEXPR reads function-definitions in DEFPROP format from all designated disk files. It also prints (<function name> REDEFINED) for any function which is redefined. DSKIN returns ***.

Example: (DSKIN SYS: SMILE ALVINE DSK: NEWFNS)

(GETDEF <device name> <file name> <list of function names>) FEXPR needs selected function definitions from specified disk file, and prints the names of those found. GETDEF returns ***.

Example: (GETDEF DSK: NEWFNS SIZE FOOBAZ)

VI. OTHER USEFUL FUNCTIONS

(TIMER <expression list>) FEXPR returns the execution time in milliseconds of the expressions in the expression list.

Example: (TIMER (GC) (GC)) returns the number of milliseconds necessary to do 2 garbage collections.

(EDIT <atom> <old> <new>) FEXPR causes all occurrences of <OLD> s-expression to be replaced by <new> s-expression in some property of <atom>. The property to change is selected as follows:

SAILON No. 41

- (1) `EXPR`
- (2) `FEXPR`
- (3) first property on property list.

Example: `(EDIT OFF TTYLENGTH 105)`

Would change OFF to:

`(DEFPROP OFF`

`(LAMBDA NIL (PROG NIL (PRINT T) (OUTC NIL T) (LINELENGTH 105)))`

`EXPR)`

`EDIT` returns T if a change was made. Otherwise NIL.

`(' <sexpr>)` MACRO is the same as `(QUOTE <sexpr>)`

`(INITF) EXPR` is a function which is useful to restore the initial states of global variables such as `linelength`, `TTY` i/o modes, etc.

`(INITFN (QUOTE INITF))` will cause `INITF` to be executed whenever bell is typed, or a LISP error return to the outer level occurs. This is useful when using `(ED)` to restore teletype input mode to `(DDTIN NIL)`.

APPENDIX I

```

(DEFPROP DE
(LAMBDA (L) (DEFIN (CAR L) (CADR L) (CADDR L) (QUOTE EXPR)))
FEXPR)

(DEFPROP DF
(LAMBDA (L) (DEFIN (CAR L) (CADR L) (CADDR L) (QUOTE FEXPR)))
FEXPR)

(DEFPROP DM
(LAMBDA (L) (DEFIN (CAR L) (CADR L) (CADDR L) (QUOTE MACRO)))
FEXPR)

(DEFPROP DV
(LAMBDA(%L)
  (PROG2 (SETQ ALLVALUES (ENTER (CAR %L) ALLVALUES))
    (SET (CAR %L) (CADR %L))))
FEXPR)

(DEFPROP ENTER
(LAMBDA (X L) (COND ((MEMBER X L) L) (T (CONS X L))))
EXPR)

(DEFPROP DEFIN
(LAMBDA(X V F P)
  (PROG (R)
    (SETQ R
      (COND
        ((GETL X (QUOTE (EXPR FEXPR SUBR FSUBR LSUBR MACRO)))
          (LIST X (QUOTE REDEFINED)))
        (T X)))
      (SETQ ALLFNS (ENTER X ALLFNS))
      (PUTPROP X
        (LIST (QUOTE LAMBDA)
          V
          (SUBST (QUOTE QUOTE) (QUOTE ') F))
        P)
      (RETURN R)))
EXPR)

```

APPENDIX II

```

(SIZE (QUOTE (COND (X Y) (T Z))))
(ENTERING 1 SIZE) ((COND (X Y) (T Z)))
(ENTERING 2 SIZE) (COND)
(LEAVING 2 SIZE) Ø
(ENTERING 2 SIZE) (((X Y) (T Z)))
(ENTERING 3 SIZE) ((X Y))
(ENTERING 4 SIZE) (X)
(LEAVING 4 SIZE) Ø
(ENTERING 4 SIZE) ((Y))
(ENTERING 5 SIZE) (Y)

```

```

(LEAVING 5 SIZE) 0
(ENTERING 5 SIZE) (NIL)
(LEAVING 5 SIZE) 0
(LEAVING 4 SIZE) 1
(LEAVING 3 SIZE) 2
(ENTERING 3 SIZE) (((T Z)))
(ENTERING 4 SIZE) ((T Z))
(ENTERING 5 SIZE) (T)
(LEAVING 5 SIZE) 0
(ENTERING 5 SIZE) ((Z))
(ENTERING 6 SIZE) (Z)
(LEAVING 6 SIZE) 0
(ENTERING 6 SIZE) (NIL)
(LEAVING 6 SIZE) 0
(LEAVING 5 SIZE) 1
(LEAVING 4 SIZE) 2
(ENTERING 4 SIZE) (NIL)
(LEAVING 4 SIZE) 0
(LEAVING 3 SIZE) 3
(LEAVING 2 SIZE) 6
(LEAVING 1 SIZE) 7

```

APPENDIX IV

```

(DEFPROP LPT
(LAMBDA NIL
 (PROG NIL (OUTC (OUTPUT LPT:) T) (OUTTIME) (LINELENGTH LPTLENGTH)))
EXPR)

```

```

(DEFPROP OFF
(LAMBDA NIL (PROG NIL (PRINT T) (OUTC NIL T) (LINELENGTH TTYLENGTH)))
EXPR)

```

```

(DEFPROP LPTOUT
(LAMBDA (L) (PROG NIL (LPT) (MAPC (FUNCTION EVAL) L) (OFF)))
FEXPR)

```

```

(DEFPROP DSKOUT
(LAMBDA(L)
 (PROG NIL
  (EVAL (LIST (QUOTE OUTPUT) (QUOTE DSK:) (CAR L)))
  (LINELENGTH LPTLENGTH)
  (OUTC T T)
  (MAPC (FUNCTION EVAL) (CDR L))
  (OFF)))
FEXPR)

```