MINI-LISP の移植*

北原栄子

1. はじめに

後藤・戸島・石畑 [1] で詳説されている MINI-LISP は，FORTRAN 言語により 784 ステップで書かれた LISP 処理系である。これは evalquote 方式の LISP であり，基本的なリスト処理と整数演算などが可能である。筆者はこれを小樽商科大学 MELCOM-COSMO (700 I, 700 S) に移植した。移植を行った目的の第1は，教材として計算機論 II の受講生に演習を行わせるためである。第2は，LISP 処理系の内容に変更を加えて新たに関数を増やすなど機能を拡張するためには FORTRAN 言語で書かれた MINI-LISP が格好であるからである。MINI-LISP 処理系の動作原理については，後藤・戸島・石畑 [1] を参照されたい。以下の記述は同書の 4. を理解していることを前提としている。以下 2. では，MINI-LISP の BATCH および TSS における起動の方法と例題を述べ，3. では移植にあたって筆者が行った処理系自体への若干の追加と変更を述べることにする。なお 3. で述べる変更を加えた MINI-LISP を本稿では商大版と呼ぶこととする。商大版は 958 ステップの FORTRAN プログラムとなっている。

2. MINI-LISP の起動

2.1 起動のための JCL

MINI-LISP のロードモジュールの名前は MINILISP である。

原稿受領日　1981年8月10日
* 本稿執筆にあたり小樽商科大学戸島熙教授より多くの有益な suggest をいただいた。記して感謝の意を表する。

2.1.1 BATCH－プログラム入力がカードの場合

```
! JOB ␣ account, name [, password]
! SET ␣ F: 55/INTLZ. P021808*; IN
! RUN ␣ (LMN, NINILISP, P021808)
! DATA
    LISP プログラム
! FIN
```

* INTLZ. P021808はディスク上に格納されている初期データのファイル名である。初期データ（商大版）の内容は附表1として本稿の終りに示した。

〔例題1〕 不定個数の引数の乗算を行う TIMES

—入力カードデック—

```
!JOB [redacted],[redacted]
!SET F:55/INTLZ.P021808;IN
!RUN (LMN,MINILISP,P021808)
!DATA
DEFINE(((TIMES(LAMBDA(X)
              (COND((NULL X) 1)(T(A* (CAR X)(TIMES (CDR X))))))) ))
TIMES((2 3 6 8 ))
TIMES((23 58 4))
!FIN
```

—出力結果—

```
                                INPUT=DEFINE(((TIMES(LAMBDA(X)
                                INPUT=              (COND((NULL X) 1)(T(A* (CAR X)(TIMES (CDR X))))))) )
 APPLY  (DEFINE (((TIMES (LAMBDA (X) (COND ((NULL X) 1) (T (A* (CAR X) (TIMES (CDR X))))))))))
   PN   1      234      5       6 6 6     78      8  7 7  8   9     9 9     A     A987654321
 =====  NIL
   PN
                                INPUT=TIMES((2 3 6 8 ))
 APPLY  (TIMES ((2 3 6 8)))
   PN   1      23       321
 =====  288
   PN
                                INPUT=TIMES((23 58 4))
 APPLY  (TIMES ((23 58 4)))
   PN   1      23       321
 =====  5336
   PN
#STOP# 0
```

2.1.2 BATCH－プログラム入力がユーザ・ファイルの場合

```
! JOB ␣ account, name [, password]
! SET ␣ F: 55/INTLZ. P021808; IN
! SET ␣ F: 5/ユーザ・ファイル名*; IN
! RUN ␣ (LMN, MINILISP, P021808)
! FIN
```

* ユーザ・ファイルは BUILD コマンドや PCL の COPY コマンドでディスク上に作られたファイルであり，EDIT プロセッサで編集可能なプログラムである。

〔例題 2〕 円盤を移動させるハノイの塔の関数 HANOI と
フィボナッチィ数 (FB) (複数ユーザ・プログラム例)

—入力カードデック—

```
!JOB [redacted],[redacted]
!SET F:S5/INTLZ.P021808:IN
!SET F:S5/HANOI:IN
!RUN (LMN*MINILISP*P021808)
!FIN
```

—出力結果—

```
                              INPUT=DEFINE(((HANOI(LAMBDA (N A B C)
                              INPUT=      (COND ((ZEROP N) NIL)(T (APPEND (HANOI (SUB1 N) A C B)
                              INPUT=            (CONS (PRINT (LIST(QUOTE EMBAN) N(QUOTE WO) A
                              INPUT=                  (QUOTE KARA) C (QUOTE E) (QUOTE UTSUSE)
                              INPUT=                  ))(HANOI (SUB1 N) B A C)))))))))))
APPLY  (DEFINE (((HANOI (LAMBDA (N A B C) (COND ((ZEROP N) NIL) (T (APPEND (HANOI (SUB1 N) A C B) (CONS (PRINT (LIST
  PN    1      234    5     6      6 6     78       8    7 7  8       9      A    A      9 9     A      B
      (QUOTE EMBAN) N (QUOTE WO) A (QUOTE KARA) C (QUOTE E) (QUOTE UTSUSE))) (HANOI (SUB1 N) B A C)))))))))
  PN    C             C   C        C   C        C   C        C C            CBA A      B      B      A987654321
=====  NIL
  PN
                              INPUT=DEFINE(((APPEND (LAMBDA (X Y)
                              INPUT=              (COND ((NULL X) Y)(T (CONS (CAR X)(APPEND (CDR X) Y)))))))))))
APPLY  (DEFINE (((APPEND (LAMBDA (X Y) (COND ((NULL X) Y) (T (CONS (CAR X) (APPEND (CDR X) Y)))))))))
  PN    1       234       5       6   6 6     78       8  7 7  8      9      9 9       A      A  987654321
=====  NIL
  PN
                              INPUT=HANOI(3 A B C)
APPLY  (HANOI (3 A B C))
  PN    1       2       21
PRINT  (EMBAN 1 WO A KARA C E UTSUSE)
  PN    1                       1
PRINT  (EMBAN 2 WO A KARA B E UTSUSE)
  PN    1                       1
PRINT  (EMBAN 1 WO C KARA B E UTSUSE)
  PN    1                       1
PRINT  (EMBAN 3 WO A KARA C E UTSUSE)
  PN    1                       1
PRINT  (EMBAN 1 WO B KARA A E UTSUSE)
  PN    1                       1
PRINT  (EMBAN 2 WO B KARA C E UTSUSE)
  PN    1                       1
PRINT  (EMBAN 1 WO A KARA C E UTSUSE)
  PN    1                       1
=====  ((EMBAN 1 WO A KARA C E UTSUSE) (EMBAN 2 WO A KARA B E UTSUSE) (EMBAN 1 WO C KARA B E UTSUSE) (EMBAN 3 WO A KA
  PN    12                      2 2                     2 2                     2 2
      RA C E UTSUSE) (EMBAN 1 WO B KARA A E UTSUSE) (EMBAN 2 WO B KARA C E UTSUSE) (EMBAN 1 WO A KARA C E UTSUSE))
  PN                  2 2                     2 2                     2 2                     21
                              INPUT=*/
```

```
                              INPUT=DEFINE(((FB(LAMBDA(N)
                              INPUT=            (COND((LTP N 2)1)
                              INPUT=            (T (A+ (FB(SUB1 N))(FB(SUB1 (SUB1 N))))))))))
APPLY  (DEFINE (((FB (LAMBDA (N) (COND ((LTP N 2) 1) (T (A+ (FB (SUB1 N)) (FB (SUB1 (SUB1 N))))))))))
  PN    1       234   5       6 6 6     78       8  7 7  8   9    A      A9 9    A      B      BA987654321
=====  NIL
  PN
                              INPUT=FB(4)
APPLY  (FB (4))
  PN    1   2 21
=====  3
  PN
                              INPUT=FB(6)
APPLY  (FB (6))
  PN    1   2 21
=====  13
  PN
*STOP* 0
```

2.1.3 TSS－端末よりプログラムを入力する場合

〔例題3〕 整数 n の階乗 $n!$ を求める関数 FC (n)

```
 MELCOM AT YOUR SERVICE -M700S
14:24 AUG 04,'81   USER# 28    LINE# 12
LOGON PLEASE:                                                    ..... ①

ON AT 14:24 AUG 04,'81
CHECK DC/MAILBOX
 EXECUTION  F10
                  ***     OTARU UNIVERSITY OF COMMERCE     ***
                  *            COMPUTER CENTER              *
                  *    MELCOM COSMO-700S TSS SERVICE TIME   *
                  *        MONDAY        9:15 - 13:15       *
                  *        TUESDAY       9:15 - 20:00       *
                  *        WEDNESDAY     9:15 - 20:00       *
                  *        THURSDAY      9:15 - 20:00       *
                  *        FRIDAY        9:15 - 20:00       *
                  *        SATURDAY      9:15 - 17:00       *
                  *    THE FIRST DAY OF A MONTH AT SEVICE   *
                  *    TIME                                 *
                  *        MONDAY       15:00 - 20:00       *
                  *      (TUE - FRI)    11:30 - 20:00       *
                  *        SATURDAY     11:30 - 17:00       *
                  ***********************************************

 *STOP* 0

!PLATEN 100

!SET F:55/INTLZ.P021808;IN                                       ..... ②

!MINILISP.P021808                                                ..... ③
 EXECUTION  F10

?MDTSS(1)                                                        ..... ④
                                    INPUT=MDTSS(1)
   APPLY  (MDTSS (1))
     PN   1      2 21
   =====  1
     PN
?MDIN(0)
                                    INPUT=MDIN(0)
   APPLY  (MDIN (0))
     PN   1     2 21
   =====  0
     PN
?DEFINE(((FC(LAMBDA(N)(COND((ZEROP N)1)(T(A* N(FC(SUB1 N))))))))))
   APPLY  (DEFINE (((FC (LAMBDA (N) (COND ((ZEROP N) 1) (T (A* N (FC (SUB1 N))))))))))
     PN   1       234   5       6 6 6     78       8  7 7  8     9   A      A987654321
   =====  NIL
     PN
?FC(7)
   APPLY  (FC (7))
     PN   1   2 21
   =====  5040
     PN
?TRACE((FC))
   APPLY  (TRACE ((FC)))
     PN   1      23  321
   =====  NIL
     PN
?FC(3)
   APPLY  (FC (3))
     PN   1   2 21
1 *APPLY  (FC (3))
     PN   1   2 21
2 *APPLY  (FC (2))
     PN   1   2 21
3 *APPLY  (FC (1))
     PN   1   2 21
4 *APPLY  (FC (0))
     PN   1   2 21
4      *=  (FC 1)
     PN   1    1
3      *=  (FC 1)
     PN   1    1
2      *=  (FC 2)
     PN   1    1
1      *=  (FC 6)
     PN   1    1
   =====  6
     PN
?
!OFF                                                             ..... ⑤

CPU = .0895 CON= 00:04:00 INT = 12 CHG = 0                       ..... ⑥
```

① account, name [,password] を入力してログオンしセッションを開いて通信可能な状態にする。

② 初期データを SET する。

③ MINILISP・P021808 を起動させ入力促進記号（プロンプト）'?' がでたらプログラム入力可能

④ TSS モードにするため MDTSS (1) を指定

⑤ LISP プログラムを終る時は ESC キーを2回押下する。

⑥ OFF 又は BYE でセッションを閉じる。

2.1.4 TSS－プログラム入力がユーザ・ファイルの場合

〔例題4〕 2つの S 式が等しければ T，等しくなければ NIL をもどす関数 EQUAL

```
 MELCOM AT YOUR SERVICE -M700S
13:42 JUL 30,'81   USER# 1D     LINE# 12
LOGON PLEASE:                                          ..... ①

ON AT 13:43 JUL 30,'81
CHECK DC/MAILBOX

!PLATEN 132

!SET F:55/INTLZ.P021808;IN                             ..... ②

!SET F:5/EQUAL;IN                                      ..... ③

!MINILISP.P021808.                                     ..... ④
 EXECUTION  F10

                                INPUT=DEFINE(((EQUAL (LAMBDA(X Y)
                                INPUT=              (COND((ATOM X)(COND((ATOM Y)(EQ X Y))(T NIL)))
                                INPUT=                    ((EQUAL(CAR X)(CAR Y))(EQUAL(CDR X)(CDR Y)))
                                INPUT=                 (T NIL))))))
   APPLY  (DEFINE (((EQUAL (LAMBDA (X Y) (COND ((ATOM X) (COND ((ATOM Y) (EQ X Y)) (T NIL))) ((EQUAL (CAR X) (CAR Y)) (E
     PN   1      234      5      6   6 6      78      8 8      9A      A A      A9 9      987 78      9      9 9      98 8
          QUAL (CDR X) (CDR Y))) (T NIL))))))))
     PN     9      9 9      987 7      7654321
   =====  NIL
     PN
                                INPUT=EQUAL((FG)(FG))
   APPLY  (EQUAL ((FG) (FG)))
     PN   1      23  3 3  321
   =====  T
     PN
                                INPUT=EQUAL((A B (C) D)(A B (C) D))
   APPLY  (EQUAL ((A B (C) D) (A B (C) D)))
     PN   1      23    4 4  3 3    4 4  321
   =====  T
     PN
                                INPUT=EQUAL(TT (K J H O))
   APPLY  (EQUAL (TT (K J H O)))
     PN   1      2   3       321
   =====  NIL
     PN
                                INPUT=EQUAL((S F H J)(S F H O))
   APPLY  (EQUAL ((S F H J) (S F H O)))
     PN   1      23       3 3       321
   =====  NIL
     PN

*STOP# 0

OFF                                                    ..... ⑥

CPU = .0915 CON= 00:02:00 INT = 9 CHG = 0
```

①, ②, ④, ⑤は 2.1.3 の①, ②, ③, ⑥に同じ。

③ユーザが作成しディスク上にファイルとしてある LISP プログラム。

2.2 XEQ コマンドの使用

MELCOM UTS/VS E01, G01 版にはカタログ JCL 化機能 (XEQ コマン

ド）がある。XEQ コマンドは，コマンドファイルと呼ばれるキードファイルまたはコンセクティブファイルからコントロールコマンドを読み込んで実行を開始するコマンドである。これによりコマンドシーケンスを簡単に実行することができるので，2.1 で述べた JCL をファイル化すると MINI-LISP の起動がより簡便になる。以下それについて述べる。

2.2.1 BATCH－プログラム入力がカードの場合

```
! JOB ␣ account, name [, password]
! XEQ* ␣ (FILE, M 5000**, P 021808)
    LISP プログラム
! FIN
```

* 700 S の場合は ! XEQ ␣ M 5000・P 021808

** ファイル名 M 5000・P 021808 の内容

```
! SET F: 55/INTLZ. P 021808; IN
! RUN (LMN, MINILISP, P 021808)
! DATA
```

〔例題 5〕 リスト x の最初の要素をみつける関数 FIRSTATM と最後の要素をみつける関数 LASTATM

—入力カードデック—

```
!JOB [redacted]
!XEQ (FILE*M5000*P021808)
DEFINE(((FIRSTATM(LAMBDA (X)
        (COND((ATOM X) X)(T(FIRSTATM (CAR X))))))
       (LASTATM (LAMBDA(X)
         (COND((NULL (CDR X))(CAR X))
           (T (LASTATM (CDR X))) ) ))
  ))
FIRSTATM(((W) Q))
LASTATM((A M E R I C ))
!FIN
```

—出力結果—

```
                    INPUT=DEFINE(((FIRSTATM(LAMBDA (X)
                    INPUT=          (COND((ATOM X) X)(T(FIRSTATM (CAR X))))))
                    INPUT=        (LASTATM (LAMBDA(X)
                    INPUT=          (COND((NULL (CDR X))(CAR X))
                    INPUT=            (T (LASTATM (CDR X))) ) ))
                    INPUT=  ))
  APPLY  (DEFINE (((FIRSTATM (LAMBDA (X) (COND ((ATOM X) X) (T (FIRSTATM (CAR X)))))) (LASTATM (LAMBDA (X) (COND ((NULL
    PN    1        234        5       6 6 6     78        8  7 7  8          9      987654 4          5        6 6 6     78
    (CDR X)) (CAR X)) (T (LASTATM (CDR X))))))))))
    PN     9       98 8      87 7  8          9      987654321
  *****  NIL
    PN
                    INPUT=FIRSTATM(((W) Q))
  APPLY  (FIRSTATM (((W) Q)))
    PN    1          234 4  321
  *****  W
    PN
                    INPUT=LASTATM((A M E R I C ))
  APPLY  (LASTATM ((A M E R I C )))
    PN    1         23           321
  *****  C
    PN
*STOP* 0
```

2.2.2 BATCH－プログラム入力がユーザ・ファイルの場合

```
! JOB ␣account, name* [, password]
! SET ␣F: 5/ユーザ・ファイル名; IN
! XEQ* ␣(FILE, M 5000 F**, P 021808)
! FIN
```

* 700 S の場合は ! XEQ ␣M 5000 F. P 021808

** ファイル名 M 5000 F. P 021808 の内容

```
! SET F: 55/INTLZ. P 021808; IN
! RUN (LMN, MINILISP, P 021808)
```

〔例題 6〕 対リストをつくる関数 PAIRLIS (700 S の使用例)

—入力カードデック—

```
!JOB [redacted],[redacted]
!SET F:5/PAIRLIS:IN
!XEQ M5000F.P021808
!FIN
```

—出力結果—

```
                              INPUT=DEFINE(((PAIRLIS(LAMBDA(LV LE LA)
                              INPUT=                 (COND((NULL LV) LA)
                              INPUT=                 (T (CONS (CONS (CAR LV)(CAR LE))
                              INPUT=                          (PAIRLIS (CDR LV)(CDR LE) LA)))))))))
 APPLY  (DEFINE (((PAIRLIS (LAMBDA (LV LE LA) (COND ((NULL LV) LA) (T (CONS (CONS (CAR LV) (CAR LE)) (PAIRLIS (CDR LV)
   PN    1      234       5          6        6 6      78        8   7 7  8      9       A       A A       A9 9         A       A
     (CDR LE) LA)))))))))
   PN    A       A   987654321
 =====  NIL
   PN
                              INPUT=PAIRLIS((RED BLUE YELLOW WHITE) (AKA AO KIIRO SHIRO)(****IROIRO NA IRO))
 APPLY  (PAIRLIS ((RED BLUE YELLOW WHITE) (AKA AO KIIRO SHIRO) (****IROIRO NA IRO)))
   PN    1        23                    3 3                   3 3                 321
 =====  ((RED . AKA) (BLUE . AO) (YELLOW . KIIRO) (WHITE . SHIRO) ****IROIRO NA IRO)
   PN    12         2 2               2 2                 2 2                   1
                              INPUT=PAIRLIS((YAMADA SATHO GOTO KIMURA)(80 50 79 100)((IKEDA.30)(ASAI.99)))
 APPLY  (PAIRLIS ((YAMADA SATHO GOTO KIMURA) (80 50 79 100) ((IKEDA . 30) (ASAI . 99))))
   PN    1        23                    3 3                   3 34                4 4                 4321
 =====  ((YAMADA . 80) (SATHO . 50) (GOTO . 79) (KIMURA . 100) (IKEDA . 30) (ASAI . 99))
   PN    12         2 2               2 2                 2 2                   2 2                 21
*STOP* 0
```

2.2.3 TSS－端末よりプログラムを入力する場合

〔例題 7〕 リスト x とリスト y を接続する関数 NCONC (700 S の使用例)

```
  MELCOM AT YOUR SERVICE -M700S
10:50 JUL 28,'81   USER# C      LINE# 12
LOGON PLEASE:                                    ..... ①

ON AT 10:50 JUL 28,'81
CHECK DC/MAILBOX
 EXECUTION  F10
                    (
                    )
!XEQ M5000T.P021808*                             ..... ②
 EXECUTION  F10
                                                 ..... ③
?MDTSS(1)                          INPUT=MDTSS(1)

   APPLY  (MDTSS (1))
     PN   1       2 21
   =====  1
     PN
?MDIN(0)                           INPUT=MDIN(0)
```

```
   APPLY  (MDIN (0))
     PN   1      2 21
   =====  0
     PN
?DEFINE(((NCONC(LAMBDA(X Y)(COND((NULL X)Y)(T(NCONC2(RPLACD(NCONC1 X)Y)X)))))
?       (NCONC1(LAMBDA(X)(COND((NULL (CDR X)) X)(T(NCONC1 (CDR X))))))
?       (NCONC2 (LAMBDA(A B) B)))))))))
   APPLY  (DEFINE (((NCONC (LAMBDA (X Y) (COND ((NULL X) Y) (T (NCONC2 (RPLACD (
     PN   1      234      5      6   6 6      78       8  7 7  8       9       A
          (COND ((NULL (CDR X)) X) (T (NCONC1 (CDR X)))))) (NCONC2 (LAMBDA (A B)
     PN   6      78       9      98  7 7  8       9      987654 4       5       6   6
   =====  NIL
     PN
?NCONC((A B C)(F H I))
   APPLY  (NCONC ((A B C) (F H I)))
     PN   1      23      3 3      321
   =====  (A B C F H I)
     PN   1          1
?NCONC((A)(V))
   APPLY  (NCONC ((A) (V)))
     PN   1      23 3 3 321
   =====  (A V)
     PN   1   1                                    ..... ④
?
!OFF                                               ..... ⑤

CPU = .6583 CON= 00:22:00 INT = 77 CHG = 0
```

①, ③, ⑤は 2.1.3 の①, ④, ⑥に同じ

② TSS での XEQ コマンドの実行

* ファイル名 M5000T. P021808 の内容

```
! SET F: 55/INTLZ. P021808; IN
! MINILISP. P021808
```

④ XEQ コマンドの終了は ESC キーを2回押下するか，又は CNTL-Y である。700I ではこの後 ***XEQ FILE INTERRUPT*** と印字され，700S ではメッセージは出さず！がでる。

2.2.4 TSS－プログラム入力がユーザ・ファイルの場合

〔例題 8〕 S-式 z の中のアトム記号 y を S-式 x でおきかえた S-式をつくる関数 SUBST

```
 WELCOM AT YOUR SERVICE -M700S
13:45 JUL 30,'81   USER# D      LINE# 12          ..... ①
LOGON PLEASE:

ON AT 13:45 JUL 30,'81
CHECK DC/MAILBOX

!PLATEN 132

!SET F:5/SUBST;IN                                  ..... ②

!XEQ M5000T.P021808                                ..... ③
 EXECUTION  F10

                                INPUT=DEFINE(((SUBST(LAMBDA(X Y Z)
                                INPUT=           (COND((ATOM Z)(COND((EQ Z Y) X)(T Z)))
                                INPUT=            (T(CONS(SUBST X Y (CAR Z))(SUBST X Y (CDR Z))))))))
                                INPUT=    )
  APPLY  (DEFINE (((SUBST (LAMBDA (X Y Z) (COND ((ATOM Z) (COND ((EQ Z Y) X) (T Z))) (T (CONS (SUBST
    PN   1      234      5      6     6 6      78       8 8      9A       A  9 9   987 7  8      9
         T X Y (CDR Z))))))))))
    PN          A      A987654321
  =====  NIL
    PN
                                INPUT=SUBST((X.A) B ((B.A).C))
  APPLY  (SUBST ((X . A) B ((B . A) . C)))
    PN   1      23      3    34      4     321
  =====  (((X . A) . A) . C)
    PN   123     3      2    1
                                INPUT=SUBST((HANAKO.SAN)  N  ((N.OHAYO) GOZAIMASU))
  APPLY  (SUBST ((HANAKO . SAN) N ((N . OHAYO) GOZAIMASU)))
    PN   1      23          3    34          4          321
  =====  (((HANAKO . SAN) . OHAYO) GOZAIMASU)
    PN   123            3          2          1

 *STOP* 0
!OFF                                               ..... ④

CPU = .2129 CON= 00:03:00 INT = 16 CHG = 0
```

①，④は 2.1.3 の①，⑥に同じ
②ユーザが作成しディスク上にある LISP プログラムファイル
③ 2.2.3 の②に同じ

3. MINI-LISP 商大版

文献 [1] の MINI-LISP を MELCOM に移植するにあたってより使い易くするためにいくつかの追加と改訂を行ったのでそれについて述べる。

3.1 セル領域の拡大

計算機内部で S-式を表現するために使われるセルの car 部と cdr 部の1次元配列 MCAR, MCDR の大きさを 1000 から 5000 に変更した。COMMON 文の MCAR (1000) を MCAR (5000) に，MCDR (994) を MCDR (4994) に，NIL=1000 を 5000 に変えてセル領域を大きくした。

3.2 I/O コードの入力

もとの MINI-LISP では 46 個の I/O コードはサブルーチン INTLZ で初期データカードの1枚目から読み込まれていたが，商大版ではあらかじめサブルーチン INTLZ の中に DATA 文で与えている。

```
      SUBROUTINE INTLZ
      COMMON/LCMCAR/MCAR(5000),ITOP,IBTTMF,JBTTMF,LRSV
      COMMON/LCMCDR/MCDR(4994),ITR,LABEL,LAMBDA,ITRUE,IFALSE,NIL
      COMMON/SYSTEM/IABASE,ICBASE,INBASE,IZERO,MARK,ICOND,IPROG
      COMMON/IOBFFR/ICHAR,INBFFR,ICGT,ICARD(80),MSSGA(7),IPRINT(110)
      COMMON /RESTART/IGCOUNT,MOIN
      COMMON/LCCODE/JARRAY(5),IOCODE(46)
      DATA  (IOCODE(J),J=1,46)/1H(,1H),1H.,1H-,1H ,1H0,1H1,1H2,1H3,
     11H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,
     21HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,
     31HX,1HY,1HZ,1H+,1H-,1H*,1H/,1H=/
```

3.3 初期データのファイル化

もとの MINI-LISP はまず初期データを入力してから各ユーザの LISP プログラムを入力する方式になっている。MINI-LISP を BATCH で使う場合複数のユーザのプログラムを一括入力する事ができるが，その時は各ユーザ・プログラムの1枚目のカードは第1欄，第2欄が */ のカードでなければならない。*/ のカードが検出されると自由領域のクリアを含む初期化がなされるので，改めて初期データの読み込みを行わなければならない。したがってユー

ザ・プログラム毎に初期データが必要になり繁雑である。商大版ではこの初期データをディスク上のファイルに格納し SET (又は ASSIGN) コマンドで DCB 'F: 55' に割りつけている。そして LISP 処理系の中では，初期データの入力とそれ以外のデータ，すなわち LISP プログラムの入力，を切り換えるために変数 MOIN を用意し，初期データ入力の時は MOIN=55 に，それ以外の時は MOIN=5 になる様にしている。こうすることによりユーザ・プログラム毎に初期データを用意する必要がなくなる。初期データの読み込みにもう1つ問題がある。それはエラーが発生した時である。MINI-LISP では BATCH の時にエラーが発生するとエラーを起したプログラムの実行を中止して */ までデータを読みとばし，次のユーザ・プログラムに対して新たに LISP プログラムの処理を再開する。しかし TSS で端末からプログラムを入力する場合にエラーが発生した時は入力促進記号 '?' の次に */ を入力すればエラー状態は回復するが，この時は制御がサブルーチン INTLZ に移るとそれまでの関数の定義が全て失われてしまうので，エラーを起こした S-式のみを無効にするとともに初期データの入力を行わない様にする必要がある。このため商大版ではモード変換関数 MDTSS (*n*) と LISP 処理系の中に変数 INTLSW を追加している。TSS の端末入力の時は MDTSS (1) としこのモードでエラーが発生した時は INTLSW=1 とした。そして INTLSW=0 の時はサブルーチン INCARD で初期データの読み込みを行い，INTLSW=1 の時はそれを行わない様にしている。

3.5 改　　頁

もとの MINI-LISP の改頁はサブルーチン LINE で行数をカウントして57行になったら改頁していたのを，商大版ではその他に第1欄が @ (アットマーク) のカードを検出したらその時も改頁する様にしている。初期データの最後のデータが@なのは，複数のユーザ・プログラムがある時，各プログラムを1頁目から出力させるためと，XEQ コマンドを使用した時コマンドファイル中のレコードの表示に続いて LISP プログラムが出力するのをさけるためである。

3.6 組込み関数の追加

表1は MINI-LISP の組込み関数一覧表である。備考欄に＊を記したものは商大版で新たに追加したもので，これについては後で述べる。

表1 MINI-LISP の組み込み関数一覧

関数名	引数の数	対応する整数	説明	備考
READ ( )	0	1	S 式を1個よみこみ値とする	
READ1 ( )	0	2	1文字よみこみ結果を文字アトムに返す	
RSERV ( )	0	3	直前の算術演算の答を 10000 で割った商（A+, A−, A＊の場合）又は余り（A/ の場合）を返す	
PRINT (*x*)	1	101	*x* を出力して終了する	
PRINT1 (*x*)	1	102	*x* を出力して終了しない	
QUOTE (*x*)	1	103	値として *x* を返す	
EVAL (*e*)	1	104	*e* の値を評価する	
ATOM (*x*)	1	105	*x* がアトムなら T，アトムでなければ F	
CAR (*x*)	1	106	*x* の car 部をとる	
CDR (*x*)	1	107	*x* の cdr 部をとる	
DEFINE (*list*)	1	108	関数定義関数	
INTP (*x*)	1	109	*x* が整数アトムなら T，そうでなければ F	
MDIN (*n*)	1	110	$n \neq 0$ なら入力カードイメージを出力 $n=0$ なら出力せず（デフォルト値は 1）	＊
ADD1 (*n*)	1	111	$n+1$	＊
SUB1 (*n*)	1	112	$n-1$	＊
ZEROP (*n*)	1	113	$n=0$ なら T，$n \neq 0$ なら F	＊
MINUS (*n*)	1	114	$0-n$	＊
ONEP (*n*)	1	115	$n=1$ なら T，$n \neq 1$ なら F	＊
LENGTH (*x*)	1	116	*x* がアトムならその印字幅，リストならその要素の数	＊
MDGB (*n*)	1	117	$n \neq 0$ ならガーベジ情報出力，$n=0$ なら出力しない（デフォルト値は 0）	＊
CAAR (*x*)	1	118	car と cdr の合成関数	＊
CADR (*x*)	1	119		＊
CDAR (*x*)	1	120		＊
CDDR (*x*)	1	121		＊
CAAAR (*x*)	1	122		＊
CAADR (*x*)	1	123		＊

関　数　名	引数の数	対応する整数	説　明	備考
CADAR (x)	1	124	car と cdr の合成関数	*
CDAAR (x)	1	125		*
CADDR (x)	1	126		*
CDDAR (x)	1	127		*
CDADR (x)	1	128		*
CDDDR (x)	1	129		*
MDTSS (n)	1	130	$n \neq 0$ のとき TSS モード，$n=0$ のとき BATCH モード（デフォルト値は 0）	*
CONS (x_1, x_2)	2	201	(x_1. x_2)	
EQ (at_1, at_2)	2	202	at_1 と at_2 が等しければ T，等しくなければ F	
APPLY (f_n, (x_1, ..., x_n))	2	203		
RPLACA (x_1, x_2)	2	204	x_1 の car 部を x_2 におきかえて値とする	
RPLACD (x_1, x_2)	2	205	x_1 の cdr 部を x_2 におきかえて値とする	
PROG2 (x_1, x_2)	2	206	引数を左から右へ評価して第2引数に値を返す	
SETQ (at, x)	2	207		
SET (x_1, x_2)	2	208	x_1 の値を x_2 の値に束縛する	
GTP (n_1, n_2)	2	209	$n_1 > n_2$ なら T，そうでなければ F	
LTP (n_1, n_2)	2	210	$n_1 < n_2$ なら T，そうでなければ F	
A+ (n_1, n_2)	2	211	$n_1 + n_2$	
A− (n_1, n_2)	2	212	$n_1 - n_2$	
A* (n_1, n_2)	2	213	$n_1 * n_2$	
A/ (n_1, n_2)	2	214	n_1 / n_2	
EXPT (n_1, n_2)	2	215	$n_1 ** n_2$	*
DIVIDE (n_1, n_2)	2	216	n_1/n_2 の商と余りを cons	*
LIST (x_1, ..., x_n)	不定	301		

記号：x…S-式，n…整数 ($-9999 \leqq n \leqq 9999$)，at…アトム記号
f_n…関数又は関数名，$list$…リスト

3.6.1 モード切り換え関数

(1) MDIN (n)

評価された n の値を変数 IMDIN に与えサブルーチン INCARD で IMDIN $\neq 0$ の時は入力カードイメージを出力させ，IMDIN$=0$ の時は出力させない様にしている。TSS の時には鍵盤入力が印字されるのでエコー出力は余計で

あるので，その時このモードを0にすればエコー出力をやめさせることができる。デフォルト値は1である。

—MDIN (*n*) 使用例—

```
?MDIN(0)
                                    INPUT=MDIN(0)
   APPLY  (MDIN (0))
     PN   1     2 21
   =====  0
     PN
?DEFINE(((NCONC(LAMBDA(X Y)
?         (COND((NULL X) Y)(T(NCONC2 (RPLACD (NCONC1 X)Y)X)))))
?        (NCONC1(LAMBDA(X)(COND((NULL (CDR X))X)(T(NCONC1(CDR X))))))
?         (NCONC2(LAMBDA(A B)  B))))))))))
   APPLY  (DEFINE (((NCONC (LAMBDA (X Y) (COND ((NULL X) Y) (T (NCONC2 (
     PN   1       234    5        6  6 6      78        8  7 7  8        9
          (COND ((NULL (CDR X)) X) (T (NCONC1 (CDR X))))))) (NCONC2 (LAMB
     PN   6      78      9      98  7 7  8        9      987654 4        5
   =====  NIL
     PN
?NCONC((A B C D)(R S T U))
   APPLY  (NCONC ((A B C D) (R S T U)))
     PN   1       23        3 3        321
   =====  (A B C D R S T U)
     PN   1                 1
?MDIN(1)
   APPLY  (MDIN (1))
     PN   1     2 21
   =====  1
     PN
?DEFINE(((FC(LAMBDA(N)
                                    INPUT=DEFINE(((FC(LAMBDA(N)
?(COND((ZEROP N)1)(T(A* N(FC(SUB1 N)))))))))
                                    INPUT=(COND((ZEROP N)1)(T(A* N(FC(SUB1
   APPLY  (DEFINE (((FC (LAMBDA (N) (COND ((ZEROP N) 1) (T (A* N (FC (SU
     PN   1       234   5        6 6 6     78        8  7 7  8     9    A
   =====  NIL
     PN

?FC(7)
                                    INPUT=FC(7)
   APPLY  (FC (7))
     PN   1   2 21
   =====  5040
     PN
?
```

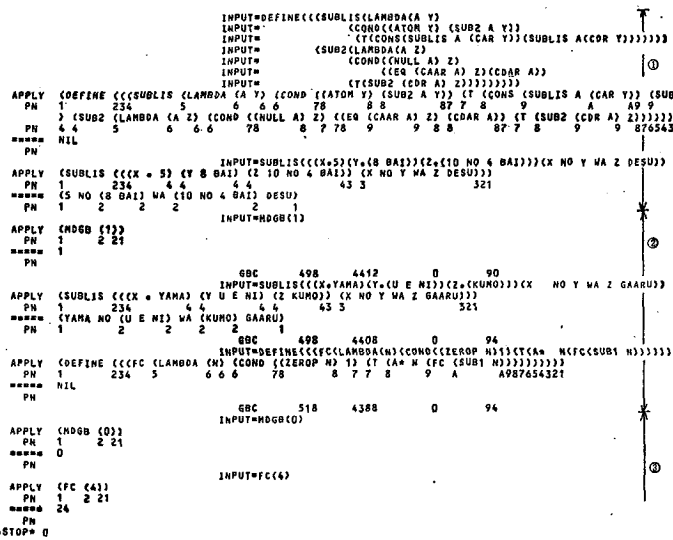
(2) MDGB (*n*)

MINI-LISP のガーベジコレクタは逆転ポインタ法を採用している。サブルーチン GBC の中で

```
      WRITE (6, 701) K1, K2, K3, K4
  701 FORMAT (38X, 3HGBC, 4I9)
```

によってガーベジ情報が出力されている。K1=IBTTMF-1（未使用領域の先頭番地-1），K2=ITOP-IBTTMF（未使用領域），K3=IABASE-ITOP（スタック領域），K4=ICBASE-IABASE（文字アトムと文字値の使用領域）をそれぞれ表わしている。商大版では関数 MDGB でこの出力の切り換えが行える様になっている。$n \neq 0$ の時はガーベジ情報を出力し $n=0$ ならガーベジ情報を出力しない。デフォルト値は0である。

— MDGB (n) 使用例 —



① デフォルト値で MDGB (n) の $n=0$。

② MDGB (1) でガーベジ情報出力モード。

③ MDGB (0) でガーベジ情報は出力されない。

(3) MDTSS (n)

前述3・4の理由で TSS であることを処理系に知らせるための関数である。TSS の時は $n=1$ とし，$n=0$ の時は BATCH である。デフォルト値は0である。TSS の端末入力の時は一番初めの入力で MDTSS (1) を指定することが必ず必要である。

— MDTSS (*n*) 使用例 —

```
?MDTSS(1)
                                    INPUT=MDTSS(1)
   APPLY  (MDTSS (1))
     PN   1      2 21
   =====  1
     PN
?MDIN(0)
                                    INPUT=MDIN(0)
   APPLY  (MDIN (0))
     PN   1      2 21
   =====  0
     PN
?DEFINE(((GET(LAMBDA(X Y)(COND((NULL X) NIL)((EQ(CAR X)Y)(CADR X))
?(T(GET(CDR X) Y))))))))
   APPLY  (DEFINE (((GET (LAMBDA (X Y) (COND ((NULL X) NIL) ((EQ (CAR X) Y) (CADR X)) (T (GET (CDR X
) Y)))))))
     PN   1       234    5       6   6 6     78      8    7 78   9     9  8 8      87 7  8    9
9  87654321
   =====  NIL
     PN
?GET((A B C D E G) B)
   APPLY  (GET ((A B C D E G) B))
     PN   1    23          3  21
   =====  C
     PN
?GET(J K O P) I)
   APPLY  (GET (J K O P))
     PN   1    2       21
                                ERROR  221
  LWLALR  (OET (COND ((NULL X) NIL) ((EQ (CAR X) Y) (CADR X)) (T (GET (CDR X) Y))) (O P))
     PN   1    2      34      4    3 34   5     5  4 4       43 3  4    5      5  432 2   21
?*/

?GET((X Y Z W) P)                               ..... ①
   APPLY  (GET ((X Y Z W) P))
     PN   1    23        3  21
   =====  NIL
     PN
?MDTSS(0)
   APPLY  (MDTSS (0))
     PN   1      2 21
   =====  0
     PN
?GET((Q W E R TY) TY)
   APPLY  (GET ((Q W E R TY) TY))
     PN   1    23          3   21
   =====  (NI . L)
     PN   1     1
?GET(((I K J L)  Q))))))))))
   APPLY  (GET (((I K J L) Q)))
     PN   1    234       4  321
                                ERROR  221
  LWLALR  (GET (LAMBDA (X Y) (COND ((NULL X) NIL) ((EQ (CAR X) Y) (CADR X)) (T (GET (CDR X) Y)))) NI
L)
     PN   1    2       3   3 3     45      5    4 45   6     6  5 5      54 4  5    6     6  5432
 1
?*/

?GET((A B C D E) B)                             ..... ②
   APPLY  (GET ((A B C D E) B))
     PN   1    23          3  21
                                ERROR  270
  LWLALR  (OET (A B C D E) B)
     PN   1    2           2  1
```

① MDTSS モードが1なので */ の後は初期化が行われないので以前に定義した関数 GET が使用出来る。

② MDTSS モードを0にしているので, */ の後は初期化が行われ関数 GET の定義も失われているので ERROR 270 (Undefined 2-argument function) のメッセージが出される。エラーが発生した時に出力されるのは ERROR 番号と LWLALR の情報であるが, LWLALR の後に出ているのは MINI-LISP のもっている register の内容に関する情報であるので, 一般利用者には特に関係がない。利用者はエラー番号をみてエラーの発生の理由を知ることができる。表2はエラーメッセージ一覧である。

表2 エラーメッセージ一覧

エラー番号	内	容
202	APPLY.	X IN LA=(X....) IS NEITHER LAMBDA NOR LABEL.
203	APPLY.	LA DEFINING A FUNCTION IS TOO SHORT.
221	APPLY.	LENGTH OF LAMBDA VARIABLE AND THAT OF ARGUMENT DO NOT MATCH.
222	APPLY.	LAMBDA VARIABLE IS NOT AN ATOM.
230	APPLY.	X IN LA=(LABEL X...) IS NOT AN ATOM.
240	APPLY.	UNDEFINED INDEFINITE-ARGUMENT FUNCTION.
243	APPLY.	UNDEFINED MORE-THAN-3-ARGUMENT FUNCTION.
250	APPLY.	UNDEFINED 0-ARGUMENT FUNCTION.
260	APPLY.	UNDEFINED 1-ARGUMENT FUNCTION.
270	APPLY.	UNDEFINED 2-ARGUMENT FUNCTION.
274	APPLY.	NON-NUMERIC ARGUMENT IN NUMERIC FUNCTION.
275	APPLY.	NUMERIC ARGUMENT IN NON-NUMERIC FUNCTION.
298	EVCON.	INVALID COND FORMAT.
400	EVCON.	ALL CONDITION FALSE IN COND.
660	SUBR.	INVALID ARGUMENT IN CAR OR CDR.
681	SUBR.	ATOM IN DEFINE LIST.
682	SUBR.	1ST ARGUMENT OF THE DEFINE PAIR IS NOT AN ATOM.
684	SUBR.	2ND ARGUMENT OF THE DEFINE PAIR IS AN ATOM.
701	GBC.	FREE STORAGE OVERFLOW.
720	SUBR.	1ST ARGUMENT OF 2-ARGUMENT FUNCTION IS INVALID.
900	INPUT.	READ ERROR.

3.6.2 算術関数

(1) ADD1 (n)

n が整数アトムかどうかを判定し，そうでなければ ERROR 274 を出力し，そうであれば $n+1$ の値を結果として戻す。

(2) SUB1 (n)

ADD1 (n) と同様に数値アトムであれば，$n-1$ を戻す。

(3) ZEROP (n)

n が整数アトムであれば $n=0$ の時 T (ITRUE の値)，$n \neq 0$ であれば F (IFALSE) の値を戻す。

(4) MINUS (n)

n が整数アトムであれば $0-n$ の値を戻す。

(5) ONEP (n)

n が整数アトムであれば $n=1$ の時 F，$n \neq 1$ の時 F を値として戻す。

(6) DIVIDE (n_1, n_2)

n_1/n_2 の商と余りを cons したものを値として戻す。

```
? DIVIDE (5432 67)
    APPLY  (DIVIDE  (5432 67))
      PN   1         2      21
     =====  (81 . 5)
      PN   1        1
```

3.6.3 その他の関数

(1) car と cdr の合成関数

caar～cdddr まで三重までの組み込み関数とした。

(2) LENGTH (x)

x が整数アトムの時は ERROR 275 を呼び，x がアトムの時はアトムの印字幅をカウントしてその値を戻し，x がリストの時はその要素の数を値として戻す関数である。

```
?LENGTH(9999)
   APPLY  (LENGTH (9999))
     PN   1        2    21
                                    ERROR  275
  LWLALR  (LENGTH 9999 (9999))
     PN   1             2    21
?LENGTH( TOKUGAWAIEYASU)
   APPLY  (LENGTH (TOKUGAWAIEYASU))
     PN   1        2              21
   =====  14
     PN
?LENGTH((HANA HATO MAME KARAKASA SAKURA))
   APPLY  (LENGTH ((HANA HATO MAME KARAKASA SAKURA)))
     PN   1        23                              321
   =====  5
```

3.6.4 組み込み関数の追加に伴うソースプログラムの変更例

前述した様な組み込み関数の追加に伴って文献[1]のソースプログラムを変更したが，それぞれの関数について述べる事は繁雑なので関数 LENGTH (x) を一例として述べる。まず初期データにあらかじめ LENGTH と対応する整数を定めておく ((LENGTH 119) 附表1. 参照)。MINI-LISP の処理系のサブルーチン LISP の中ではこの整数は変数 IBF に assign される。第113行の1引数関数かどうかを判定する IF 文

```
IF (IBF-100) 262, 8260, 8260
```

の100を新たに定義した1引数の対応する整数の最も大きい数+1（商大版では131）とする。次に第115行の GO TO 文

GO TO (630, 630, 640, 300, 650, 660, 670, 680, 690), ICGT

を ICGT=16 (IBF-100) である時の jump 先を指定する様に変更する。すなわち

GO TO (630, 630, 640, 300, 650, 660, 670, 680, 690, ..., **697,** ...), ICGT

とする。そして関数 LENGTH の処理を行う statement no. 697 の以下のプログラムを第233行の後につけると，LISP プログラムの中で関数 LENGTH を使用する事ができる。

```
C     *SUBROUTINE LENGTH(LARG)
  697 IF (LARG-INBASE) 6987,8275,8275
 6987 LL=IZERO
      MCALARG=MCAR(LARG)
      GO TO 6971
 8275 CALL ERROR(275)
C 275 APPLY.   NUMERIC ARGUMENT IN NON-NUMERIC FUNCTION.
 6973 LL=LL+1
      LARG=MCDR(LARG)
 6971 IF(LARG-NIL)    6972,6974,6975
 6972 IF (MCAR(LARG)-NIL) 6973,6973,6976
 6974 LARG=LL
      GO TO 640
 6975 LARG=MCALARG
 6977 IF (MCDR(LARG)-NIL) 6970,6970,6979
 6970 LL=LL+2
      LARG=MCDR(LARG)
      GO TO 6977
 6979 LL=LL+2
      GO TO 6980
 6976 LL=LL+1
      LARG=MCAR(LARG)
      GO TO 6981
 6980 LARG=MCDR(LARG)
 6981 LLM=LARG-5001
      MOLLM=MOD(LLM,47)
      IF(MOLLM) 6978,6974,6978
 6978 LL=LL+1
      GO TO 6974
```

〈参 考 文 献〉

[1] 後藤英一・戸島　熙・石畑　清「記号処理の基礎と応用」（情報処理叢書8），情報処理学会，1981年。

[2] 中西正和「LISP 入門—システムとプログラミング—」（コンピュータサイエンス大学講座），近代科学社，1977年。

```
A014:   KITAHARA,¥¥¥¥¥¥¥    07/20/81    15:31

 NAME = INTLZ

     1 -     1.000  301 (NIL F T LAMBDA LABEL TRFLAG COND QUOTE SETQ PROG2 AND OR F1 F2 F3 PROG)
     2 -     2.000   108 (((READ 1) (READ1 2) (RSERV 3) (PRINT 101) (PRINT1 102) (QUOTE 103) (EVAL
     3 -     3.000    104) (ATOM 105) (CAR 106) (CDR 107) (DEFINE 108) (INTP 109) (MDIN 110)
     4 -     4.000  (ADD1 111) (SUB1 112) (ZEROP 113) (MINUS 114) (ONEP 115) (LENGTH 116)
     5 -     5.000  (MDGB 117)
     6 -     6.000  (CAAR 118) (CADR 119)( CDAR 120)(CDDR 121)
     7 -     7.000  (CAAAR 122) (CAADR 123) (CADAR 124) (CDAAR 125)
     8 -     8.000  (CADDR 126) (CDDAR 127) (CDADR 128)(CDDDR 129) (MDTSS 130)
     9 -     9.000  (CONS 201) (EQ 202)
    10 -    10.000  (APPLY 203) (RPLACA 204) (RPLACD 205) (PROG2 206) (SETQ 207) (SET 208)
    11 -    11.000    (GTP 209) (LTP 210) (A+ 211) (A- 212) (A# 213) (A/ 214)
    12 -    12.000,  (EXPT 215) (DIVIDE 216)                (LIST 301)
    13 -    13.000  ))
    14 -    14.000  DEFINE ((
    15 -    15.000 (PROG (LAMBDA #U ((LAMBDA (#V) (COND ((ATOM (SETQ #W (CAR #U))) (#PROG))
    16 -    16.000              (T (#PV1)))) #U)))
    17 -    17.000 (#PV1 (LAMBDA () (APPLY (LIST (QUOTE LAMBDA) (LIST (CAR #W))
    18 -    18.000       (QUOTE (COND ((SETQ #W (CDR #W)) (#PV1)) (T (#PROG))))) (LIST NIL))))
    19 -    19.000 (#PROG (LAMBDA () (COND
    20 -    20.000       ((ATOM (SETQ #V (CDR #V))) NIL)
    21 -    21.000       ((ATOM (SETQ #W (CAR #V))) (#PROG))
    22 -    22.000       (T (#PEVAL)))))
    23 -    23.000 (#PEVAL (LAMBDA () (COND
    24 -    24.000       ((EQ (SETQ #CARW (CAR #W)) (QUOTE COND)) (#PCOND))
    25 -    25.000       ((EQ #CARW (QUOTE RETURN)) (COND ((ATOM (SETQ #W (CDR #W))) NIL)
    26 -    26.000            (T (EVAL (CAR #W)))))
    27 -    27.000       ((EQ #CARW (QUOTE GO)) (PROG2 (LIST (SETQ #W (CAR (CDR #W)))
    28 -    28.000            (SETQ #V #U)) (#GO)))
    29 -    29.000       ((LIST (EVAL #W)) (#PROG)))))
    30 -    30.000 (#PCOND (LAMBDA () (COND
    31 -    31.000       ((ATOM (SETQ #W (CDR #W))) (#PROG))
    32 -    32.000       ((ATOM (SETQ #CARW (CAR #W))) (PRINT (QUOTE (ERROR IN #PCOND))))
    33 -    33.000       (((LAMBDA (#W) (EVAL (CAR #CARW))) #W) (COND
    34 -    34.000            ((ATOM (SETQ #W (CAR (CDR (CAR #W))))) (#PROG)) (T (#PEVAL))))
    35 -    35.000       (T (#PCOND)))))
    36 -    36.000 (#GO (LAMBDA () (COND
    37 -    37.000       ((ATOM (SETQ #V (CDR #V))) NIL)
    38 -    38.000       ((EQ #W (CAR #V)) (#PROG))
    39 -    39.000       (T (#GO)))))
    40 -    40.000 ))
    41 -    41.000 DEFINE ((
    42 -    42.000 (TRACE (LAMBDA (TLIST) (COND
    43 -    43.000       ((ATOM TLIST)  NIL)
    44 -    44.000       (T (PROG2 (SET (CAR TLIST)  (CONS TRFLAG (104  (CAR TLIST))))
    45 -    45.000                 (       TRACE (CDR TLIST))))
    46 -    46.000 )))
    47 -    47.000 (UNTRACE (LAMBDA (TLIST) (COND
    48 -    48.000       ((ATOM TLIST)  NIL)
    49 -    49.000       (T (PROG2 (SET (CAR TLIST)  (CDR (104  (CAR TLIST))))
    50 -    50.000                 (       UNTRACE (CDR TLIST))))
    51 -    51.000 )))
    52 -    52.000 ))
    53 -    53.000 DEFINE ((
    54 -    54.000       (NULL (LAMBDA (X) (EQ X NIL)))
    55 -    55.000 ))
    56 -    56.000 a
```

附表1　初期データリスト（商大版）