

[/ PROVE] TRACK 36
CREATED 15.50 2 11 1973 NEW

[16.25 2 NOV 1973]

COMMENT 'THIS FILE CONTAINS THE TOP-LEVEL THEOREM PROVER, "PROVE".
"PROVE1" DOES ALL THE WORK. MOST OF THE FUNCTIONS
BETWEEN HERE AND THERE ARE CONCERNED ONLY WITH OUTPUT.';

VARS THM THMTIME STUCKTHM AVOIDSTARS VERBOSE LASTPPRTHM REPORTFN;
1->AVOIDSTARS;
0->VERBOSE;
NIL->SPECPROF;

COMMENT 'THIS RECOGNIZES WHEN THE THEOREM HAS BEEN BEATEN TO DEATH.';
FUNCTION FINISHED TERM;

VARS FUNSYM;
IF ATOM(TERM)
THEN 1;
ELSEIF (HD(TERM)->FUNSYM;FUNSYM="EQUAL")
OR FUNSYM="CAR" OR FUNSYM="CDR" OR FUNSYM="CONS"
THEN
LOOPIF (TL(TERM)->TERM;TERM/=NIL)
THEN
IF FINISHED(HD(TERM)) THEN; ELSE 0;EXIT;
CLOSE;
1;
ELSE 0;CLOSE;
END;

FUNCTION GENSKOLIST LIST;
MAPLIST(LIST,LAMBDA CONST;CONSPAIR(CONST,GENSKO(CONST));END);
END;

COMMENT 'THIS FUNCTION APPLIES FERTILIZATION AND IF THAT FAILS
TRIES GENERALIZING AND INDUCTING. IT IS CAREFUL TO WORK ONLY
ON THE FIRST CONJUNCT IF THE THEOREM IS A CONJUNCT. FOR THIS
IT GETS THE NAME "ARTIFICIAL INTELLIGENCE", BEING ABOUT THE
SMARTEST PROGRAM IN THE THEOREM PROVER.';

FUNCTION ARTIFINTEL THM;
IF FERTILIZE(THM)
THEN
1;
EXIT;
0->CONJFLAG;
IF HD(THM)="IF"
THEN
IF HD(TL(TL(THM)))=NIL
THEN
[% "IF", [% "NOT", HD(TL(THM)) %], HD(TL(TL(TL(THM)))) , NIL %]->THM;
1->CONJFLAG;
ELSEIF HD(TL(TL(TL(THM))))=NIL
THEN
1->CONJFLAG;

```

        CLOSE;
    CLOSE;
IF CONJFLAG
    THEN
        HD(TL(THM));(REPORT("&",'(WORK ON FIRST CONJUNCT ONLY)',"ARTIFINTEL"));
    ELSE
        THM;
    CLOSE;
->INDTERM;
GENERALIZE(INDTERM)->INDTERM;
IF INDUCT(INDTERM)
    THEN
        ->INDTERM;
    ELSE INDTERM;0;EXIT;
IF CONJFLAG
    THEN
        [%"IF",INDTERM,
        APPSUBST(GENSKOLIST(INDCONSTS),HD(TL(TL(THM))))),
        NIL%];
    ELSE
        INDTERM;
    CLOSE;
1;
END;

```

COMMENT THIS IS THE THEOREM PROVER. ASTOUNDING IN ITS SIMPLICITY.
 THE OUTPUT FUNCTIONS HAVE BEEN MOVED TO THE SIDE TO REVEAL THE
 ESSENCE OF THE SYSTEM: BEAT THE THEOREM TO DEATH WITH
 EVALUATION, NORMALIZE AND REDUCE. IF THAT FAILS, TRY A LITTLE
 AI AND THEN MORE VIOLENCE.'

```

FUNCTION PROVE1 THM;
SETUP();(REPORT(NIL,'THEOREM TO BE PROVED:',"PROVE1"));
MACEX(THM)->THM;
LOOP:(REPORT("/",IDENTFN,"PROVE1"));
THM->OLDTHM;
SYMEVAL(THM)->THM;(REPORTIF("E",'EVALUATION YIELDS:',"PROVE1",LASTREPTHM));
NORMALIZE(THM)->THM;(REPORTIF("N",'WHICH NORMALIZES TO:',"PROVE1",LASTREPTHM));
REDUCE(THM)->THM;(REPORTIF("R",'AND REDUCES TO:',"PROVE1",LASTREPTHM));
IF FINISHED(THM)
    THEN (REPORTIF(NIL,'WHICH IS EQUIVALENT TO:',
        "PROVE1",LASTPPRTHM));
    (REPORT(".",FINREPORT,"PROVE1"));

    WRAPUP();
    EXIT;
IF EQUAL(THM,OLDTHM)
    THEN
        (REPORTIF(NIL,'WHICH IS EQUIVALENT TO:',
            "PROVE1",LASTPPRTHM));
    IF ARTIFINTEL(THM)
        THEN
            ->THM;(REPORT("","THE THEOREM TO BE PROVED IS NOW:',
                "PROVE1"));
        ELSE
            ->THM;
            THM->STUCKTHM;(REPORT("Q",'STUCK',"PROVE1"));

```

```

    FINREPORT();WRAPUP();
    EXIT;
  CLOSE;
GOTO LOOP;
END;

```

COMMENT 'THE TOP-LEVEL. MAINLY CONCERNED WITH I/O, LIKE
 RECOGNIZING WHEN THE USER WANTS ALL THE THEOREMS IN THE STANDARD
 FILE PROVED, OR WHEN HE HAS GIVEN YOU A THEOREM NAME RATHER THAN
 A THEOREM.';

```

FUNCTION PROVE THM;
VARS TOTTIME;
IF HD(THM)="ALL"
  THEN
    0->TOTTIME;
    POPTON();NL(10);PR(PODATE());NL(4);
    APPLIST(GETTHM(THM),
      LAMBDA THM;
        IF AVOIDSTAR AND MEMBER("*,HD(THM)) THEN EXIT;
        PROVE1(THM);TOTTIME+THMTIME->TOTTIME;END);
    POPTON();
    NL(10);
    PRSTRING('TOTAL TIME: ');PR(TOTTIME);PRSTRING(' SECS.');
```

```

    NL(5);
  ELSEIF ISNUMBER(HD(THM))
    THEN
      PROVE1([%THM,GETTHM(THM)%]);
    ELSE PROVE1(THM);CLOSE;
END;

```

```

FUNCTION LPPROVE LIST;
VARS LPLNFEEDS DDF2 PPRMAXLNS MARG2 THMNAME;
180->PPRMAXLNS;
79->MARG2;
POPMESS([LP80 THEOREMS PROVED])->DDF2;
IF HD(LIST)="ALL" THEN GETTHM(LIST)->LIST;CLOSE;
APPLIST(LIST,
  LAMBDA X;
    CHAROUT->CUCHAROUT;
    0->LPLNFEEDS;
    NL(2);
    IF NOT(ATOM(HD(X))) THEN HD(X); ELSE NIL;CLOSE;
    ->THMNAME;
    IF AVOIDSTAR AND MEMBER("*,THMNAME) THEN EXIT;
    PR(THMNAME);
    NL(1);
    LAMBDA X;
    DDF2(X);
    IF X=17
      THEN

```

```

LPLNFEEDS+1->LPLNFEEDS;
IF LPLNFEEDS=60
  THEN
    2->LPLNFEEDS;
    SP(60); IF THMNAME/=NIL THEN PR(THMNAME);CLOSE;DDF2(17);DDF2(17);
    CLOSE;
  CLOSE;
END->CUCHAROUT;
DDF2(64);
IF VERBOSE=0 THEN 0.5->VERBOSE;CLOSE;
PROVE(X);
END;);
CHAROUT->CUCHAROUT;
DDF2(TERMIN)
END;

```

```

FUNCTION SIMPLIFY TERM;
VARS OLDTERM;
UNDEF->OLDTERM;
LOOPIF NOT(EQUAL(OLDTERM,TERM))
  THEN
    TERM->OLDTERM;
    REDUCE(NORMALIZE(SYMEVAL(TERM)))->TERM;
    CLOSE;
  TERM;
END;

```

```

FUNCTION READLOOP;
LOOPIF TRUE
  THEN
    PPR(RLOOPFN(MACEX(READ())));NL(2);
    CLOSE;
  END;

```

```

SYMEVAL->RLOOPFN;

```

[THEOREMS] TRACK 36
CREATED 16.57 1 11 1973

[16.25 2 NOV 1973]

[[1 0 * COMMENT]
['THEOREMS ABOUT APPEND, REVERSE AND LENGTH']]

[[1 1]
[EQUAL [APPEND A [APPEND B C]]
[APPEND [APPEND A B] C]]]

[[1 2]
[IMPLIES [EQUAL [APPEND A B] [APPEND A C]]
[EQUAL B C]]]

[[1 3]
[EQUAL [LENGTH [APPEND A B]]
[LENGTH [APPEND B A]]]]]

[[1 4]
[EQUAL [REVERSE [APPEND A B]]
[APPEND [REVERSE B] [REVERSE A]]]]]

[[1 5]
[EQUAL [LENGTH [REVERSE A]]
[LENGTH A]]]

[[1 6]
[EQUAL [REVERSE [REVERSE A]] A]]]

[[1 7]
[IMPLIES A
[EQUAL [LAST [REVERSE A]] [CAR A]]]]]

[[1 8]
[IMPLIES [EVEN I]
[EQUAL A [REVN I A]]]]]

[[2 0 * COMMENT]
['THEOREMS INVOLVING MEMBER']]

[[2 1]
[IMPLIES [MEMBER X A]
[MEMBER X [APPEND B A]]]]]

[[2 2]
[IMPLIES [MEMBER X A]
[MEMBER X [APPEND A B]]]]]

[[2 3]
[IMPLIES [OR [MEMBER X A] [MEMBER X B]]
[MEMBER X [APPEND A B]]]]]

[[2 4]
[IMPLIES [AND [MEMBER X A] [MEMBER X B]]
[MEMBER X [INTERSEC A B]]]]]

[[2 5]
[IMPLIES [OR [MEMBER X A] [MEMBER X B]]]

```

    [MEMBER X [UNION A B]]]
[[2 6]
 [IMPLIES [SUBSET A B]
 [EQUAL [UNION A B] B]]]
[[2 7]
 [IMPLIES [SUBSET A B]
 [EQUAL [INTERSEC A B] A]]]
[[2 8]
 [EQUAL [MEMBER X A]
 [NOT [EQUAL [ASSOC X [PAIRLIST A B]]
 NIL]]]]
[[3 0 * COMMENT]
 ['THEOREMS INVOLVING MAPLIST']]
[[3 1]
 [EQUAL [MAPLIST [APPEND A B] X]
 [APPEND [MAPLIST A X] [MAPLIST B X]]]]
[[3 2]
 [EQUAL [LENGTH [MAPLIST A X]]
 [LENGTH A]]]
[[3 3]
 [EQUAL [MAPLIST [REVERSE A] X]
 [REVERSE [MAPLIST A X]]]]
[[4 0 * COMMENT]
 ['THEOREMS INVOLVING MISC FUNCTIONS']]
[[4 1]
 [IMPLIES [AND [BOOLEAN X] [BOOLEAN Y]]
 [EQUAL [AND [IMPLIES X Y] [IMPLIES Y X]]
 [EQUAL X Y]]]]
[[4 2]
 [EQUAL [ELEMENT I A]
 [ELEMENT [PLUS [LENGTH C] I]
 [APPEND C A]]]]
[[4 3 * UNPROVEN]
 [IMPLIES [ELEMENT I A]
 [MEMBER [ELEMENT I A] A]]]
[[4 4]
 [EQUAL [EQUAL X Y] [EQUAL Y X]]]
[[4 5]
 [IMPLIES [AND [EQUAL X Y] [EQUAL Y Z]]
 [EQUAL X Z]]]
[[4 6]
 [IMPLIES [AND [BOOLEAN X]
 [AND [BOOLEAN Y] [BOOLEAN Z]]]
 [EQUAL [EQUAL X [EQUAL Y Z]]
 [EQUAL [EQUAL X Y] Z]]]]

```

[[4 7]
 [EQUAL [ELEMENT I A]
 [CAR [CDRN I A]]]]

[[4 8]
 [IMPLIES [ELEMENT I A]
 [LTE I [LENGTH A]]]]

[[5 0 * COMMENT]
 ['THEOREMS INVOLVING ARITHMETIC']]

[[5 1] [EQUAL [PLUS I J] [PLUS J I]]]

[[5 2]
 [EQUAL [PLUS I [PLUS J K]]
 [PLUS [PLUS I J] K]]]

[[5 3]
 [EQUAL [PLUS [PLUS I J] K]
 [PLUS [PLUS J K] I]]]

[[5 4]
 [EQUAL [TIMES I J] [TIMES J I]]]

[[5 5]
 [EQUAL [TIMES I [PLUS J K]]
 [PLUS [TIMES I J] [TIMES I K]]]]]

[[5 6]
 [EQUAL [TIMES I [TIMES J K]]
 [TIMES [TIMES I J] K]]]

[[5 7] [EVEN [DOUBLE I]]]

[[5 8] [EQUAL [HALF [DOUBLE I]] I]]]

[[5 9]
 [IMPLIES [EVEN I]
 [EQUAL [DOUBLE [HALF I]] I]]]

[[5 10]
 [EQUAL [DOUBLE I] [TIMES 2 I]]]

[[5 11]
 [EQUAL [DOUBLE I] [TIMES I 2]]]

[[5 12] [EQUAL [EVEN I] [EVEN2 I]]]

[[6 0 * COMMENT]
 ['THEOREMS INVOLVING ORDERING RELATIONS']]

[[6 1]
 [GT [LENGTH [CONS A B]] [LENGTH B]]]

[[6 2]
 [IMPLIES [AND [GT I J] [GT J K]]
 [GT I K]]]

[[6 3]
 [IMPLIES [GT I J] [NOT [GT J I]]]]]

[[6 4] [LTE I [PLUS J I]]]
[[6 5] [OR [LTE I J] [LTE J I]]]
[[6 6]
 [OR [GT I J]
 [OR [GT J I] [EQUAL I J]]]]
[[6 7] [ORDERED [SORT A]]]
[[6 8]
 [EQUAL [MEMBER X [SORT A]]
 [MEMBER X A]]]
[[6 9]
 [EQUAL [LENGTH A] [LENGTH [SORT A]]]]
[[6 10 * LONG]
 [EQUAL [COUNT X A]
 [COUNT X [SORT A]]]]
[[6 11]
 [IMPLIES [ORDERED A]
 [EQUAL A [SORT A]]]]
[[6 12]
 [IMPLIES [ORDERED [APPEND A B]]
 [ORDERED A]]]
[[6 13 * UNPROVEN]
 [IMPLIES [ORDERED [APPEND A B]]
 [ORDERED B]]]
[[6 14 * LONG]
 [EQUAL [EQUAL [SORT A] A]
 [ORDERED A]]]
[[6 15] [LTE [HALF I] I]]
[[6 16]
 [IMPLIES [AND [ORDERED A]
 [AND [ORDERED B]
 [LTE [LAST A] [CAR B]]]]
 [ORDERED [APPEND A B]]]]
[[7 0 * COMMENT]
 ['THEOREMS INVOLVING TREE STRUCTURES']]
[[7 1] [EQUAL [COPY X] X]]
[[7 2] [EQUAL [SUBST X X Y] Y]]
[[7 4]
 [IMPLIES [NOT [OCCUR X Y]]
 [EQUAL [SUBST Z X Y] Y]]]
[[7 5 * UNPROVEN]
 [IMPLIES [NOT [OCCUR NIL X]]
 [EQUAL [SWAPTREE [SWAPTREE X] X]]]]


```
[[7 6 * UNPROVEN]
 [IMPLIES [NOT [OCCUR NIL X]]
  [EQUAL [FLATTEN [SWAPTREE X]]
   [REVERSE [FLATTEN X]]]]]
```

```
[[7 7]
 [EQUAL [LENGTH [FLATTEN X]]
  [TIPCOUNT X]]]
```

```
[[8 0 * COMMENT]
 ['THEOREMS ABOUT BINARY ARITHMETIC']]
```

```
[[8 1 * UNPROVEN]
 [EQUAL [BINARYOF [PLUS I J]]
  [BINADD [BINARYOF I] [BINARYOF J]]]]]
```

```
[[8 2]
 [EQUAL [NUMOFBIN [BINARYOF I]] I]]]
```

```
[[8 3]
 [EQUAL [NUMOFBIN [CDR [BINARYOF I]]]
  [HALF I]]]
```

[/ SORTFILE] TRACK 36
CREATED 16.54 1 11 1973

[16.26 2 NOV 1973]

COMPILE(LIBRARY([ALLSORT]));

```
FUNCTION WORNCOMP X Y;  
IF ISWORD(X)  
  THEN IF ISWORD(Y) THEN ALFER(X,Y); ELSE 0;CLOSE;  
ELSEIF ISWORD(Y) THEN 1; ELSE X<Y;CLOSE;  
END;
```

```
FUNCTION KEYCOMP X Y;  
IF ATOM(X)  
  THEN IF ATOM(Y) THEN WORNCOMP(X,Y); ELSE 1;CLOSE;  
ELSEIF ATOM(Y) THEN 0;  
  ELSE  
  LOOP:  
  IF X=NIL THEN 1;EXIT;  
  IF Y=NIL THEN 0;EXIT;  
  IF HD(X)=HD(Y) THEN TL(X)->X;TL(Y)->Y;GOTO LOOP;CLOSE;  
  WORNCOMP(HD(X),HD(Y));  
  CLOSE;  
END;
```

```
FUNCTION SORTFILE TRACK FILE;  
VARS MARG2 CUCHAROUT PPRSP F001 PPRSTRING;  
DTRACK(TRACK);  
HD(TL(TL(DDFIND(FILE))))->F001;  
IF F001+DISCEND>159 THEN PRSTRING('DISC TOO FULL');SETPOP();  
  CLOSE;  
ALLSORT([%APPFILE(FILE,IDENTFN)%],LAMBDA X Y;  
  KEYCOMP(HD(X),HD(Y));END)->F001;
```

```
79->MARG2;  
SP->PPRSP;  
PR->PPRSTRING;  
DOUT(FILE)->DDF2;  
DDF2->CUCHAROUT;  
APPLIST(F001,LAMBDA;PPR();NL(2);END);  
DDF2(TERMIN);  
END;
```

```
PRSTRING('SORTFILE(<TRK>,<FILE>) => ( ). ');NL(2);
```

[/ DEFS] TRACK 36
CREATED 16.53 1 11 1973

[16.26 2 NOV 1973]

[ADD1 [LAMBDA [X] [CONS NIL X]]]

[ADDTOLIS [LAMBDA [X Y]
[COND [[NULL Y] [CONS X NIL]]
[[LTE X [CAR Y]] [CONS X Y]]
[T [CONS [CAR Y]
[ADDTOLIS X [CDR Y]]]]]]]

[AND [LAMBDA [X Y] [IF X Y NIL]]]

[APPEND [LAMBDA [X Y]
[IF X
[CONS [CAR X] [APPEND [CDR X] Y]]
Y]]]

[ASSOC [LAMBDA [X Y]
[COND [[NULL Y] NIL]
[[EQUAL X [CAAR Y]] [CAR Y]]
[T [ASSOC X [CDR Y]]]]]

[ATOM [LAMBDA [X] [IF X [EQUAL [CAR X] NIL] T]]]

[BINADD [LAMBDA [X Y]
[COND [[NULL X] Y]
[[NULL Y] X]
[[ZEROP [CAR X]]
[CONS [CAR Y]
[BINADD [CDR X] [CDR Y]]]]
[[ZEROP [CAR Y]]
[CONS 1 [BINADD [CDR X] [CDR Y]]]]
[T [CONS 0
[BINADD [QUOTE [1]]
[BINADD [CDR X] [CDR Y]]]]]]]

[BINARYOF [LAMBDA [X]
[IF [ZEROP X]
NIL
[BINADD [QUOTE [1]]
[BINARYOF [SUB1 X]]]]]]]

[BOOLEAN [LAMBDA [X] [OR [EQUAL X T] [EQUAL X NIL]]]]

[CAAR [LAMBDA [X] [CAR [CAR X]]]]

[CADR [LAMBDA [X] [CAR [CDR X]]]]

[CDAR [LAMBDA [X] [CDR [CAR X]]]]

[CDDR [LAMBDA [X] [CDR [CDR X]]]]

[CDRN [LAMBDA [X Y]
[IF [ZEROP X]
Y
[CDRN [SUB1 X] [CDR Y]]]]]

```

[ COPY [LAMBDA [X]
  [IF [ATOM X]
    X
    [CONS [COPY [CAR X]]
          [COPY [CDR X]]]]]]
]

[ COUNT [LAMBDA [X Y]
  [COND [[NULL Y] 0]
        [[EQUAL X [CAR Y]]
         [ADD1 [COUNT X [CDR Y]]]]
        [T [COUNT X [CDR Y]]]]]
]

[ DOUBLE [LAMBDA [X]
  [IF [ZEROP X]
    0
    [ADD1 [ADD1 [DOUBLE [SUB1 X]]]]]]]
]

[ ELEMENT [LAMBDA [X Y]
  [IF [ZEROP X] [CAR Y] [ELEMENT [SUB1 X] [CDR Y]]]]]
]

[ EVEN [LAMBDA [X] [IF [ZEROP X] T [ODD [SUB1 X]]]]]
]

[ EVEN2 [LAMBDA [X]
  [COND [[ZEROP X] T]
        [[ZEROP [SUB1 X]] NIL]
        [T [EVEN2 [SUB1 [SUB1 X]]]]]]]
]

[ EXP [LAMBDA [X Y]
  [IF [ZEROP Y]
    1
    [TIMES X [EXP X [SUB1 Y]]]]]
]

[ FLATTEN [LAMBDA [X]
  [IF [ATOM X]
    [LIST X]
    [APPEND [FLATTEN [CAR X]]
            [FLATTEN [CDR X]]]]]
]

[ FNSEGS [LAMBDA [X]
  [IF [NULL X]
    NIL
    [CONS X [FNSEGS [CDR X]]]]]
]

[ GT [LAMBDA [X Y]
  [COND [[ZEROP X] NIL]
        [[ZEROP Y] T]
        [T [GT [SUB1 X] [SUB1 Y]]]]]
]

[ HALF [LAMBDA [X]
  [COND [[ZEROP X] 0]
        [[ZEROP [SUB1 X]] 0]
        [T [ADD1 [HALF [SUB1 [SUB1 X]]]]]]]
]

[ IMPLIES [LAMBDA [X Y] [IF X Y T]]]
]

[ INTERSEC [LAMBDA [X Y]
  [COND [[NULL X] NIL]
        [[MEMBER [CAR X] Y]
         [CONS [CAR X] [INTERSEC [CDR X] Y]]]
        [T [INTERSEC [CDR X] Y]]]]]
]

```

```

[ISBINARY [LAMBDA [X]
  [COND [[ATOM X] [NULL X]]
        [[OR [EQUAL [CAR X] 0]
              [EQUAL [CAR X] 1]]
        [ISBINARY [CDR X]]]]]]

[LAST [LAMBDA [X]
  [IF [NULL [CDR X]]
      [CAR X]
      [LAST [CDR X]]]]]

[LENGTH [LAMBDA [X]
  [IF [NULL X]
      0
      [ADD1 [LENGTH [CDR X]]]]]]

[LTE [LAMBDA [X Y]
  [COND [[ZEROP X] T]
        [[ZEROP Y] NIL]
        [T [LTE [SUB1 X] [SUB1 Y]]]]]]

[MAPLIST [LAMBDA [X Y]
  [IF X
      [CONS [APPLY Y [CAR X]]
            [MAPLIST [CDR X] Y]]
      NIL]]]

[MEMBER [LAMBDA [X Y]
  [COND [[NULL Y] NIL]
        [[EQUAL X [CAR Y]] T]
        [T [MEMBER X [CDR Y]]]]]]

[NOT [LAMBDA [X] [IF X NIL T]]]

[NULL [LAMBDA [X] [EQUAL X NIL]]]

[NUMBERP [LAMBDA [X]
  [IF X
      [IF [CAR X] NIL [NUMBERP [CDR X]]]
      T]]]

[NUMOFBIN [LAMBDA [X]
  [COND [[NULL X] 0]
        [[ZEROP [CAR X]]
         [DOUBLE [NUMOFBIN [CDR X]]]]
        [T [ADD1 [DOUBLE [NUMOFBIN [CDR X]]]]]]]]

[OCCUR [LAMBDA [X Y]
  [COND [[EQUAL X Y] T]
        [[ATOM Y] NIL]
        [T [OR [OCCUR X [CAR Y]]
                [OCCUR X [CDR Y]]]]]]]]

[ODD [LAMBDA [X]
  [IF [ZEROP X] NIL [EVEN [SUB1 X]]]]]

[OR [LAMBDA [X Y] [IF X T Y]]]

[ORDERED [LAMBDA [X]

```

```

[COND [[NULL X] T]
      [[NULL [CDR X]] T]
      [[LTE [CAR X] [CADR X]]
       [ORDERED [CDR X]]]]]]

[PAIRLIST [LAMBDA [X Y]
          [IF [NULL X]
              NIL
              [CONS [CONS [CAR X] [CAR Y]]
                     [PAIRLIST [CDR X] [CDR Y]]]]]]]]

[PLUS [LAMBDA [X Y]
      [IF [ZEROP X]
          Y
          [ADD1 [PLUS [SUB1 X] Y]]]]]]

[REVERSE [LAMBDA [X]
         [IF [NULL X]
             NIL
             [APPEND [REVERSE [CDR X]]
                      [LIST [CAR X]]]]]]]]

[REVN [LAMBDA [X Y]
      [IF [ZEROP X]
          Y
          [REVERSE [REVN [SUB1 X] Y]]]]]]

[SORT [LAMBDA [X]
      [IF [NULL X]
          NIL
          [ADDTOLIS [CAR X] [SORT [CDR X]]]]]]]]

[SUB1 [LAMBDA [X] [CDR X]]]

[SUBSET [LAMBDA [X Y]
        [COND [[NULL X] T]
              [[MEMBER [CAR X] Y]
               [SUBSET [CDR X] Y]]]]]]]]

[SUBST [LAMBDA [X Y Z]
      [COND [[EQUAL Y Z] X]
            [[ATOM Z] Z]
            [T [CONS [SUBST X Y [CAR Z]]
                     [SUBST X Y [CDR Z]]]]]]]]]]

[SWAPTREE [LAMBDA [X]
         [IF [ATOM X]
             X
             [CONS [SWAPTREE [CDR X]]
                    [SWAPTREE [CAR X]]]]]]]]

[TIMES [LAMBDA [X Y]
      [IF [ZEROP X]
          0
          [PLUS Y [TIMES [SUB1 X] Y]]]]]]

[TIPCOUNT [LAMBDA [X]
         [IF [ATOM X]
             1
             [PLUS [TIPCOUNT [CAR X]]]]]]]]

```

[TIPCOUNT [CDR X]]]]]]

[UNION [LAMBDA [X Y]

[COND [[NULL X] Y]

[[MEMBER [CAR X] Y] [UNION [CDR X] Y]]

[T [CONS [CAR X] [UNION [CDR X] Y]]]]]]]

[ZEROP [LAMBDA [X] [EQUAL X 0]]]

[/ VERBOSE] TRACK 36
CREATED 16.52 1 11 1973

[16.26 2 NOV 1973]

```
FUNCTION PRSEQ NLCOUNT STR LIST PRFN CONNECTOR;
IF VERBOSE
  THEN
    POPTON();
    NL(NLCOUNT);
    PRSTRING(STR);
    LOOP:
    PRFN(HD(LIST));
    TL(LIST)->LIST;
    IF LIST=NIL THEN PRSTRING(',');NL(2);EXIT;
    IF TL(LIST)=NIL
      THEN PRSTRING(CONNECTOR);
      ELSE PRSTRING(', ');
      CLOSE;
    GOTO LOOP;
  CLOSE;
END;

PRSEQ('%' AND '%')->PRSEQAND;
PRSEQ('%' OR '%')->PRSEQOR;

IDENTFN->REPORTFN;

FUNCTION REPORT CODE CMT BRKCMT;
VARS X;
IF CODE=NIL THEN GOTO VERB;CLOSE;
IF CHAROUT/=CUCHAROUT OR NOT(VERBOSE)
  THEN
    CUCHAROUT;CHAROUT->CUCHAROUT;PR(CODE);->CUCHAROUT;
    CLOSE;
CODE::PROFILE->PROFILE;
REPORTFN();
VERB:
THM->LASTREPTHM;
IF ISFUNC(CMT) THEN CMT();GOTO CHKSPEC;CLOSE;
IF VERBOSE=1 OR VERBOSE=0.5 AND CODE/="E" AND CODE/="N"
  AND CODE/="R"

  THEN
    POPTON();NL(4);PRSTRING(CMT);NL(2);
    IF SUBSRC(DATALength(CMT),CMT)=10 THEN
      THM->LASTPPRTHM;PPR(THM);NL(2);CLOSE;
    CLOSE;
CHKSPEC:
IF CODE=NIL THEN EXIT;
IF SPECPROF/=NIL
  THEN
    IF EQUAL(CODE,HD(SPECPROF))
      THEN
        TL(SPECPROF)->SPECPROF;
        IF HD(SPECPROF)="+"
          THEN
            HD(TL(SPECPROF))->X;
```



```

        TL(TL(SPECPROF))->SPECPROF;
        CUCHAROUT;CHAROUT->CUCHAROUT;
        POPTTON();NL(4);PRSTRING('INTERRUPT: ');PR(BRKCMT);NL(1);->CUCHAROUT;
        IF ISLINK(X)
            THEN POPVAL(X<>[; GOON]);
            ELSE APPLY(VALOF(X));CLOSE;
        CUCHAROUT;CHAROUT->CUCHAROUT;NL(1);PRSTRING('RESUMED');NL(4);->CUCHAROU
T;
        CLOSE;
    ELSE
        CHAROUT->CUCHAROUT;POPTTON();NL(1);PRSTRING('PROFILES DIVERGE');NL(1);SETP
OP();
        CLOSE;
    CLOSE;
END;

```

```

FUNCTION REPORTIF TESTTHM;
IF EQUAL(THM,TESTTHM) THEN ERASE();ERASE();ERASE();EXIT;
REPORT();
END;

```

```

FUNCTION SETUP;
0->ININDUCT;
NIL->PROFILE;
0->STARCOUNT;
NIL->STARALIST;
NIL->GENRLALIST;
NIL->BOMBLIST;
NIL->PROVEFNS;
NIL->GENALIST;
NIL->INDLIST;
UNDEF->ENDTHM;
POPTIME->THMTIME;
IF NOT(ATOM(HD(THM)))
    THEN
        NL(2);
        PR(HD(THM));
        SP(5);
        HD(TL(THM))->THM;
        CLOSE;
THM->LASTREPTHM;
THM->LASTPPRTHM;
IF VERBOSE
    THEN PR(POPDATE());CLOSE;
NL(1);
END;

```

```

FUNCTION FINREPORT;
POPTTON();
IF VERBOSE=1
    THEN
        IF PROVEFNS/=NIL
            THEN
                NL(5);PRSTRING('FUNCTION DEFINITIONS:');NL(2);
                APPLIST(REV(PROVEFNS),
                    LAMBDA FN;
                    IF FN="*" THEN EXIT;
                    PPR([%FN,PROP("DEFN",FN)%]);
                    IF PROP("PROGGEND",FN)=1

```

```

        THEN NL(1);PRSTRING('(PROGRAM GENERATED)');CLOSE;
        NL(2);
        END);
    CLOSE;
IF STARALIST/=NIL
    THEN
    NL(3);PRSTRING('HIDDEN TERMS:');NL(2);
    APPLIST(REV(STARALIST),
        LAMBDA X;
        PR(FRONT(X));PRSTRING(' = ');PPRIND(BACK(X),5,0);NL(2);
        END);
    CLOSE;
IF GENRLALIST/=NIL
    THEN
    NL(3);PRSTRING('GENERALIZATIONS:');NL(2);
    APPLIST(REV(GENRLALIST),
        LAMBDA X;
        PR(FRONT(X));PRSTRING(' = ');PR(BACK(X));NL(2);
        END);
    CLOSE;
CLOSE;
IF VERBOSE
    THEN
    NL(3);
    PRSTRING('PROFILE: ');PR(REV(PROFILE));NL(2);
    CLOSE;
END;

```

```

FUNCTION WRAPUP;
THM->ENDTHM;
(POPTIME-THMTIME)/16->THMTIME;
IF NOT(VERBOSE) OR CHAROUT/=CUCHAROUT
    THEN
    CUCHAROUT;CHAROUT->CUCHAROUT;
    NL(1);PPR(THM);NL(1);PR(THMTIME);NL(1);
    ->CUCHAROUT;
    CLOSE;
IF VERBOSE
    THEN
    NL(4);
    PRSTRING('TIME: ');PR(THMTIME);PRSTRING(' SECS. ');
    NL(5);
    CLOSE;
END;

```

COMMENT THEOREMS DEEMED NOT IMPORTANT ENOUGH TO PROVE
ALL THE TIME, OR THEOREMS WHICH EXPLOIT A TRICK
IN OUR LISP NOT IN OTHERS (LIKE NUMBERP);

[T 0 1]::
[IMPLIES [AND [NOT [EQUAL A B]] [MEMBER A [CONS B [CONS C NIL]]]]
[MEMBER A [CONS C NIL]]];

[T 0 2]::
[IMPLIES [NUMBERP A] [EQUAL [LENGTH A] A]];

[T 0 3]::
[EQUAL [LENGTH [LENGTH A]] [LENGTH A]];

[T 0 4]::
[EQUAL [LENGTH [APPEND A B]] [APPEND [LENGTH A] [LENGTH B]]];

[T 0 5]::
[NUMBERP [LENGTH A]];

[T 0 6]::
[EQUAL [LENGTH [REVERSE A]] [REVERSE [LENGTH A]]];

[T 0 7]::
[IMPLIES [AND [NUMBERP A] [NUMBERP B]] [EQUAL [APPEND A B] [APPEND B A]]];

[T 0 8]::
[IMPLIES [EQUAL [LENGTH A] [CONS NIL B]] [EQUAL [LENGTH [CDR A]] B]];

[T 0 9]::
[IMPLIES [NUMBERP A] [EQUAL [REVERSE A] A]];

[T 0 10]::
[IMPLIES [GT A B] [NOT [EQUAL A B]]];

[T 0 11]::
[IMPLIES A [GT [APPEND A B] B]];

[T 0 12]::
[IMPLIES [AND [ORDERED A] [LTE B [CAR A]]]
[ORDERED [CONS B A]]];

[T 0 13]::
[IMPLIES [AND A [ORDERED A]] [ORDERED [CDR A]]];

[QUANT DEFS] TRACK 36
CREATED 16.50 1 11 1973

[16.27 2 NOV 1973]

FUNCTION MAC X1;
HD(TL(X1))->PROP("MAC",HD(X1));
END;

[ALL [IF LIST [IF [SCH [CAR LIST]] [ALL] NIL] T]].MAC;
[SOME [IF LIST [IF [SCH [CAR LIST]] T [SOME]] NIL]].MAC;
[NUM [IF LIST [IF [SCH [CAR LIST]] [CONS NIL[NUM]] [NUM]] NIL]].MAC;
[MAPLIST[IF LIST [CONS [SCH [CAR LIST]] [MAP]] NIL]].MAC;
[MIN [IF LIST [IF [MIN] [MIN]] [IF [SCH LIST] [CONS NIL LIST] NIL]]
NIL]].MAC;
[MAX [IF LIST [IF [SCH LIST] [CONS NIL LIST] [MAX]] NIL]].MAC;
[SET [IF LIST [IF [SCH [CAR LIST]] [CONS [CAR LIST] [SET]] [SET]] NIL]]
.MAC;

[QUANTIFY] TRACK 36
CREATED 16.50 1 11 1973

[16.27 2 NOV 1973]

```
FUNCTION DEQUANTIFY L;  
VARS DEF MAC FORMVAR ACCVAL BODY VLIST NEWNAME;  
IF ATOM(L) THEN L;EXIT;  
PROP("MAC",HD(L))->DEF;  
IF DEF=UNDEF THEN MAPLIST(L,DEQUANTIFY);EXIT;  
HD(L)->MAC;  
HD(TL(L))->FORMVAR;  
HD(TL(TL(L)))->ACCVAL;  
HD(TL(TL(TL(L))))->BODY;  
DEQUANTIFY(BODY)->BODY;  
[%FORMVAR%]->VLIST;  
ALLV(BODY);  
REV(VLIST)->VLIST;  
GENSYM(MAC,0)->NEWNAME;  
SCHEMATIZE(DEF)->DEF;
```

```
DEFINE([%NEWNAME,[%"LAMBDA",VLIST,DEF%]]);  
MAC->PROP("MACFUN",NEWNAME);  
BODY->PROP("BODY",NEWNAME);  
NEWNAME:::(ACCVAL:::TL(VLIST));  
END;
```

```
FUNCTION ALLV L;  
IF ATOM(L) THEN  
  IF CONSTANT(L) OR MEMBER(L,VLIST) THEN  
    ELSE CONS(L,VLIST)->VLIST;CLOSE;  
  ELSE APPLIST(TL(L),ALLV);  
  CLOSE;  
END;
```

```
FUNCTION SCHEMATIZE L;  
IF ATOM(L) THEN IF CONSTANT(L) THEN L; ELSE FORMVAR;CLOSE;EXIT;  
IF HD(L)=MAC THEN NEWNAME:::([%"CDR",FORMVAR%]::TL(VLIST));EXIT;  
IF HD(L)="SCH" THEN SUBST(SUBST(FORMVAR,"LIST",HD(TL(L))),FORMVAR,  
  BODY);EXIT;  
HD(DEF):::MAPLIST(TL(L),SCHEMATIZE);  
END;
```

```
FUNCTION CONSTANT L;  
IF L=NIL OR L="T" OR ISNUMBER(L) THEN 1; ELSE 0;CLOSE;  
END;
```

```
FUNCTION REQUANTIFY L;  
VARS BODY VALS;  
IF ATOM(L) THEN L;EXIT;  
PROP("MACFUN",HD(L))->MAC;  
IF MAC=UNDEF THEN MAPLIST(L,REQUANTIFY);EXIT;  
PROP("BODY",HD(L))->BODY;  
TL(TL(L))->VALS;  
APPLIST(TL(HD(TL(PROP("DEFN",HD(L))))),  
  LAMBDA X1;SUBST(HD(VALS),X1,BODY)->BODY;TL(VALS)->VALS;END);
```

```
[%MAC,HD(HD(TL(PROP("DEFN",HD(L))))),  
HD(TL(L)),REQUANTIFY(BODY)%];  
END;
```

```
VARS OPPR NPPR;  
PPR->OPPR;  
FUNCTION NPPR X1;  
IF ATOM(X1) THEN X1.PR;  
ELSEIF SHD(SHD(TL(X1)))="LAMBDA" THEN X1.OPPR;  
ELSE OPPR(REQUANTIFY(X1));CLOSE;  
END;
```

[MONTHMS] TRACK 36
CREATED 16.49 1 11 1973

[16.27 2 NOV 1973]

[T N 1]::
[EQUAL [APPEND A B] [APPEND B A]];

[T N 2]::
[EQUAL [REVERSE [APPEND A B]] [APPEND [REVERSE A] [REVERSE B]]];

[T N 3]::
[EQUAL [LENGTH [REVERSE A]] A];

[T N 4]::
[EQUAL [REVERSE A] A];

[T N 5]::
[IMPLIES [MEMBER A B] [MEMBER A [INTERSECT B C]]];

[T N 6]::
[IMPLIES [MEMBER A B] [MEMBER A [INTERSECT C B]]];

[T N 7]::
[EQUAL [REVERSE [MAPLIST A B]] [MAPLIST A B]];

[T N 8]::
[IMPLIES [NUMBERP A] [EVEN A]];

[T N 9]::
[EQUAL [ADD A B] [MULT A B]];

[T N 10]::
[EQUAL [MULT [CONS NIL A] B] [MULT A [CONS NIL B]]];

[T N 11]::
[GT A A];

[T N 12]::
[IMPLIES [ORDERED A] [ORDERED [REVERSE A]]];

[T N 13]::
[IMPLIES [GT 2 [LENGTH A]] [BOOLEAN A]];

[T N 14]::
[GT 3 [LENGTH A]];

[T N 15]::
[IMPLIES [AND A [ORDERED A]] [EQUAL [MAX A] [LAST A]]];

[. DLP80] TRACK 36
CREATED 16.41 1 11 1973

[16.27 2 NOV 1973]

```
FUNCTION DLP801;
VARS CUCHAROUT DDG2 DDG3;
DDF2->CUCHAROUT;1->DDG3;
L:2.NL;
LO:.DDF1->DDG2;
IF DDG2=63
  THEN 64.DDF2;67.SP;"PAGE".PR;PRINT(DDG3+1)->DDG3;GOTO L CLOSE;
IF DDG2=TERMIN THEN EXIT;
DDG2.DDF2;
GOTO LO;
END;
LAMBDA X1;
VARS DDMPFLAG OLDTRACK DDF1 DDF2 CUCHAROUT X2 DDG1;
0->DDMPFLAG;
DISCUSER->OLDTRACK;
TOP:
IF ISLINK(X1)
  THEN
    IF ISLINK(HD(X1))
      THEN
        POPMESS([[LP80 15] FILE DUMP])->DDF2;
        DDF2->CUCHAROUT;
        NL(2);
        APPLIST(X1,LAMBDA;PR();NL(2);END);

        APPLIST(X1,
          LAMBDA X1;
          DDF2(64);
          DDFIND(X1)->DDG3;
          IF DDG3 THEN ELSE CHAROUT->CUCHAROUT;
            X1.PR,"Q".PR;DDF2->CUCHAROUT;EXIT;DISC(X1)->DDF1;
          PR(X1);SP(2);PR("TRACK");SP(1);PR(DISCUSER);SP(35);
          PR(POPDATE());NL(1);"CREATED".PR,1.SP,DDG3.TL.TL.TL.DDATEPR;
          NL(2);
          DLP801();
          IF LENGTH(DDG3)>5 THEN 1->DDMPFLAG;
            NIL->DDG3.TL.TL.TL.TL.TL;CLOSE;
          END);
        DDF2(TERMIN);
      ELSE
        DDFIND(X1)->DDG3;
        IF DDG3 THEN ELSE "Q".PR;EXIT;

        POPMESS("LP80":X1)->DDF2;
        DDF2->CUCHAROUT;
        "TRACK".PR;DISCUSER.PR;SP(2);"CREATED".PR,1.SP;
        DDG3.TL.TL.TL.DDATEPR;
        NL(2);
        DISC(X1)->DDF1;
        DLP801();
        DDF2(TERMIN);
        IF LENGTH(DDG3)>5 THEN 1->DDMPFLAG;
          NIL->DDG3.TL.TL.TL.TL.TL;CLOSE;
        CLOSE;
      ELSE

```



```

IF ISINTEGER(X1)
  THEN 100000->X2;
  ELSE INTOF(X1);INTOF((X1-INTOF(X1)+0.01)*10)->X2->X1;CLOSE;
DTRACK(X1);
DISCDIR->DDG1;
IF X2 THEN
  ([% LOOPIF DDG1/=NIL AND X2
    THEN IF HD(HD(DDG1))/="FREE" THEN HD(HD(DDG1));
        X2-1->X2;CLOSE;TL(DDG1)->DDG1;CLOSE%])
  ELSE
  [% LOOPIF DDG1/=NIL
    THEN
    IF LENGTH(HD(DDG1))>5 AND HD(HD(DDG1))/="FREE"
      THEN HD(HD(DDG1));CLOSE;
    DDG1.TL->DDG1;
    CLOSE%];
  CLOSE;
->X1;
GOTO TOP;
CLOSE;
IF DDMPFLAG THEN DDMP();CLOSE;
DTRACK(OLDTRACK);
END.APPLY;

0->DLP801;

```

[. FILEFT] TRACK 36
CREATED 12.32 24 10 1973

[16.28 2 NOV 1973]

```
OPERATION 2 FILEFT FILE TRK1 TRK2;  
IF ISLINK(HD(FILE))  
  THEN  
  DTRACK(TRK2);  
  APPLIST(FILE,LAMBDA X1;  
    IF DDFIND(X1) THEN DKILL(X1); CLOSE; END);  
  IF ISFUNC(EDSETPOP) THEN VALOF("EDTIDY").APPLY;  
  ELSE DTIDY(); CLOSE;  
  APPLIST(FILE,LAMBDA X;FILEFT(X,TRK1,TRK2);END); EXIT;  
  DREPIN(FILE,DTRACK(TRK1),DIN(FILE),DTRACK(TRK2));  
  END;  
FILEFT();
```