21.16HRS. 14 SEPT 1973. ** RSB **

[FILE DUMP]

[ADDEFS]

[/ INPUT]

[/ EVAL]

[/ GEN]

[/ TYPE]

[/ SORTDEFS]

[/ GENSYM]

[/ PROPS]

[PPR]

[/ IDENT]

[ALAN]

9

THE THEOREM PROVER
AS REPORTED IN
J'S THESIS.

[ADDEFS] TRACK 9 [21.16 14 SEPT 1973]
CREATED 20.03 14 9 1973

```
FUNCTION ADDEFS X;
VARS CUCHAROUT U V;

DIN([/DEFS])->U;
DOUT([/DEFS])->CUCHAROUT;
APPLIST(X,LAMBDA X;
PRSTRING('DEFINE
(`);
PPRIND([% X, PROP("DEFN",X) %], 1 , 2);
PRSTRING(');
`);
END);
DDIO(U,CUCHAROUT);
END;
ADDEFS();
```

```
[/ INPUT]  TRACK  9                                        [ 21.16 14 SEPT 1973]
CREATED    20.03   14  9 1973


COMMENT `THIS FILE CREATES THE FUNCTION "DEFINE" WHICH JUST PUTS
A FUNCTION DEFINITION ON THE PROPERTY LIST OF THE FUNCTION NAME.
THE FUNCTION "GETTHM" JUST LINKS TO OUR DISC TRACK TO FETCH
A THEOREM FROM THE STANDARD THEOREM FILE, GIVEN A THEOREM NAME.
(THIS IS NOT AN ESSENTIAL PART OF THE PROGRAM.)`;

VARS THMTRACK THMFILE;
36 -> THMTRACK;
[THEOREMS] -> THMFILE;

NIL -> ALLFNS;

FUNCTION DEFINE X;
[% "DEFN", HD(TL(X)) %] -> MEANING(HD(X));
IF MEMBER(HD(X),ALLFNS)
 THEN; ELSE HD(X) :: ALLFNS -> ALLFNS; CLOSE;
END;
FUNCTION NORMDEF X;
VARS PROVEFNS LEXPR FNNAME;
IF ISWORD(X)
 THEN X; PROP("DEFN",X);
 ELSE HD(X); HD(TL(X)); CLOSE;
-> LEXPR -> FNNAME;
NIL -> PROVEFNS;
NORMALATE(HD(TL(TL(LEXPR))))->X;
[% LOOPIF PROVEFNS /= NIL
  THEN IF HD(PROVEFNS) /= FNNAME AND
     PROP("DEFN",HD(PROVEFNS))=UNDEF
    THEN HD(PROVEFNS); CLOSE;
  TL(PROVEFNS) -> PROVEFNS;
  CLOSE %] -> PROVEFNS;
IF PROVEFNS /= NIL
 THEN NL(2); PR(FNNAME);NL(1);
 PRSTRING('UNDEFINED FUNCTIONS:  ');
 PR(PROVEFNS);
 NL(2);
 CLOSE;
DEFINE([% FNNAME, [% "LAMBDA", HD(TL(LEXPR)), X %] %]);
END;



FUNCTION GETTHM THMNAME;
VARS PROGLIST TRK;
DISCUSER->TRK;
DTRACK(THMTRACK);
IF HD(THMNAME) = "ALL"
 THEN [% COMPILE(DIN(THMFILE)) %]; DTRACK(TRK);
   -> TRK;
   IF TL(THMNAME) /= NIL
    THEN
    HD(TL(THMNAME)) -> THMNAME;
    LOOPIF NOT(EQUAL(THMNAME,HD(HD(TRK)))) THEN TL(TRK)->TRK;CLOSE;
```

```
    CLOSE;
   TRK;
     EXIT;
INCHARITEM(DIN(THMFILE)) -> LIST;
DTRACK(TRK);
FNTOLIST(LAMBDA;LOOP:LIST()->TRK;IF TRK="COMMENT"
   THEN LOOPIF LIST()/=";" THEN CLOSE; GOTO LOOP; CLOSE;
   TRK;END) -> PROGLIST;
LOOPIF NOT(NULL(PROGLIST))
 THEN
 IF EQUAL(THMNAME,LISTREAD()) THEN ERASE(ITEMREAD());LISTREAD();EXIT;
 LOOPIF ITEMREAD() /= ";" THEN CLOSE;
 CLOSE;
ERRFUN(THMNAME,57);
END;

FUNCTION ADDEFS;

DCOMP([ADDEFS]);
END;
```

[/ EVAL] TRACK 9
CREATED   15.27   15 6 1973

[21.16 14 SEPT 1973]

```
VARS PROVEFNS APPLYNONPRIM BOMBOUT OTHERFAILS POCKETIT BOMBED EXPNDGFUN
     BINDVARS AUXANALY ANALYSIS;
NIL -> PROVEFNS;

COMMENT 'THIS IS THE BASIC EVAL ROUTINE. ';

FUNCTION EVAL TERM;
VARS Y X;

COMMENT 'IF TERM IS ATOM, RETURN IT OR VALUE, ACCORDING
TO WHICH ATOM.';

IF ATOM(TERM)
 THEN
 IF TERM = NIL OR TERM = "T" OR ISNUMBER(TERM)
  THEN TERM;
 ELSEIF ASSOC(TERM,ALIST)
  THEN BACK();
  ELSE TERM; CLOSE;
 EXIT;

COMMENT 'GET FUNCTION SYMBOL';

HD(TERM) -> X;

COMMENT 'CONSIDER THE POSSIBILITIES';

IF X = "CAR"
 THEN
 POCKETIT -> X;
 0 -> POCKETIT;
 EVAL(HD(TL(TERM))) -> Y;
 X -> POCKETIT;
 IF Y = NIL
  THEN NIL;
 ELSEIF Y = "T"
  THEN NIL;
 ELSEIF ISNUMBER(Y)
  THEN NIL;
 ELSEIF ISNUMSKO(Y)
 THEN NIL;
 ELSEIF SHD(Y) = "CONS"
  THEN HD(TL(Y));
  ELSE BOMBOUT([% "CAR", Y %]);
  EXIT;
 STEPCNT + 1 -> STEPCNT;
 EXIT;

IF X = "CDR"
 THEN
 POCKETIT -> X;
 0 -> POCKETIT;
 EVAL(HD(TL(TERM))) -> Y;
```

```
 X -> POCKETIT;
 IF Y = NIL
 THEN NIL;
 ELSEIF Y = "T"
  THEN NIL;
 ELSEIF ISNUMBER(Y)
  THEN Y - 1;
 ELSEIF SHD(Y) = "CONS"
  THEN HD(TL(TL(Y)));
  ELSE BOMBOUT([% "CDR", Y %]);
  EXIT;
 STEPCNT + 1 -> STEPCNT;
 EXIT;

IF X = "CONS"
 THEN
 COMMENT '[% "CONS", EVAL(HD(TL(TERM))), EVAL(HD(TL(TL(TERM)))) %]';
EVAL(HD(TL(TERM)))->X;
EVAL(HD(TL(TL(TERM))))->Y;
[% "CONS", X , Y %];

 EXIT;

IF X = "EQUAL"
 THEN
 EVAL(HD(TL(TERM))) -> X;
 EVAL(HD(TL(TL(TERM)))) -> Y;
 IDENT(Y,X) -> FOO1;
 IF FOO1 = NIL
  THEN NIL;
 ELSEIF FOO1
  THEN "T";
 ELSEIF ISCONS(Y) AND ISCONS(X)
  THEN
  APPLY(ALIST,LAMBDA ALIST;
  CONSPAIR("X",X) :: (CONSPAIR("Y",Y) :: ALIST) -> ALIST;
EVAL([COND [EQUAL [CAR X] [CAR Y]] [EQUAL [CDR X] [CDR Y]] NIL]) ;
 END);RETURN;
  STEPCNT + 1 -> STEPCNT;
  GOTO CONDRULES;
  ELSE
  [% "EQUAL", X, Y %];
  EXIT;
 STEPCNT + 1 -> STEPCNT;
 EXIT;

IF X = "COND"
 THEN
CONDRULES:
 EVAL(HD(TL(TERM))) -> Y;
 IF Y = NIL OR Y = 0
  THEN
  EVAL(HD(TL(TL(TL(TERM)))));
 ELSEIF ISCONS(Y)
  THEN
  EVAL(HD(TL(TL(TERM))));
  ELSE
  COMMENT '[%"COND",Y,EVAL(HD(TL(TL(TERM)))),
EVAL(HD(TL(TL(TL(TERM)))))%]';
EVAL(HD(TL(TL(TERM))))->X;
```

```
EVAL(HD(TL(TL(TL(TERM)))))->FOO1;
[% "COND", Y, X, FOO1 %];
EXIT;
STEPCNT+1->STEPCNT;
EXIT;

COMMENT 'X MUST BE NON-PRIMITIVE, CAREFULLY EVAL IT';
APPLYNONPRIM();

END;


FUNCTION BOMBOUT TERM;
IF POCKETIT
 THEN
 TERM :: POCKET -> POCKET;
 ELSE TERM :: OTHERFAILS -> OTHERFAILS;
 CLOSE;
TERM;
END;

FUNCTION EVALARGS;
VARS POCKET POCKETIT;
NIL -> POCKET;
(HD(TERM) = EXPNDGFUN) -> POCKETIT;
MAPLIST(TL(TERM),EVAL);
IF POCKET /= NIL
 THEN
 POCKET :: BOMBLIST -> BOMBLIST;
 1 -> BOMBED;
 CLOSE;
END

FUNCTION EXPANDCALL EXPNDGFUN ALIST;
IF NOT(MEMBER(EXPNDGFUN,PROVEFNS))
 THEN EXPNDGFUN :: PROVEFNS -> PROVEFNS; CLOSE;
PROP("DEFN",EXPNDGFUN) -> FOO1;
IF FOO1 = UNDEF
 THEN
 EXPNDGFUN :: EVALDARGS;
 EXIT;
NIL -> BOMBLIST;
NIL -> OTHERFAILS;
BINDVARS(EVALDARGS,HD(TL(FOO1)),ALIST)->ALIST;
EVAL(HD(TL(TL(FOO1))));
END;

FUNCTION BINDVARS ARGLIST VARLIST ALIST;
LOOPIF ARGLIST /= NIL
 THEN
 CONSPAIR(HD(VARLIST),HD(ARGLIST)) :: ALIST -> ALIST;
 TL(ARGLIST) -> ARGLIST;
 TL(VARLIST) -> VARLIST;
 CLOSE;
ALIST;
END;

FUNCTION APPLYNONPRIM;
VARS EVALDARGS SAVEANALY SAVEBLIST SAVEOTHER SAVEPOCK;
EVALARGS() -> EVALDARGS;
```

```
IF BOMBED THEN HD(TERM) :: EVALDARGS; EXIT;
ANALYSIS -> SAVEANALY;
BOMBLIST -> SAVEBLIST;
OTHERFAILS -> SAVEOTHER;
POCKET->SAVEPOCK;
EXPANDCALL(HD(TERM),ALIST) ->F001;
IF BOMBED
 THEN
 HD(TERM) :: EVALDARGS;
SAVEPOCK->POCKET;
 [% "***", HD(TERM), BOMBLIST, OTHERFAILS,HD(TERM),"***" %]
:: SAVEANALY -> ANALYSIS;
 ELSE
 F001;
 SAVEANALY -> ANALYSIS;
 STEPCNT + 1 -> STEPCNT;
 CLOSE;
0 -> BOMBED;
SAVEBLIST -> BOMBLIST;
SAVEOTHER -> OTHERFAILS;
END;



FUNCTION EVALUATE TERM;
VARS ALIST;
NIL->ALIST;
NIL -> BOMBLIST;
NIL -> ANALYSIS;
NIL -> OTHERFAILS;
UNDEF -> EXPNDGFUN;
0 -> POCKETIT;
0 -> BOMBED;
0 -> STEPCNT;
EVAL(TERM);
END;
```

```
[/ GEN]  TRACK  9                                   [ 21.16 14 SEPT 1973]
CREATED   15.05    1  6 1973


COMMENT 'THIS FILE SUPPLIES MANY LIST PROCESSING FUNCTIONS THAT
SHOULD BE STANDARD TO POP-2.  IN ADDITION, SEVERAL FUNCTIONS
FOR RECOGNIZING CERTAIN CLASSES OF LISP EXPRESSIONS ARE PROVIDED.
THESE INCLUDE SKOLEM CONSTANTS, EXPRESSIONS COMPOSED ONLY OF
"CONS" AND "NIL" (CALLED REALLINKS IN THIS PROGRAM), EXPRESSIONS
THAT START WITH "CONS" AND EXPRESSIONS STARTING WITH ANY LISP
PRIMITIVE.';

VARS ASSOCID MEMBERID XAPPFLAG VERBOSE F001 F002 F003 IDENT PROP
     ASSOC MEMBER GENSYM;

FUNCTION GENMEM X L EQFN;
LOOPIF L /= NIL
 THEN IF EQFN(X,HD(L)) THEN 1; EXIT;
 TL(L) -> L;
 CLOSE;
0;
END;

GENMEM(% EQ %) -> MEMBER;

FUNCTION NCONC L1 L2;
IF L1 = NIL
 THEN L2;
 ELSE L1;
 LOOPIF (TL(L1) /= NIL) THEN TL(L1) -> L1;CLOSE;
 L2 -> TL(L1);
 CLOSE;
END;

FUNCTION DELETE X L;
VARS L0;
L -> L0;
IF L = NIL THEN NIL;
ELSEIF HD(L) = X THEN TL(L);
 ELSE
LOOP:
 IF TL(L) = NIL THEN L0; EXIT;
 IF HD(TL(L)) = X THEN TL(TL(L)) -> TL(L); L0; EXIT;
 TL(L) -> L;
 GOTO LOOP;
 CLOSE;
END;

FUNCTION XAPPLIST L FN;
0 -> XAPPFLAG;
LOOPIF L /= NIL AND NOT(XAPPFLAG)
 THEN
 FN(HD(L));
 TL(L) -> L;
 CLOSE;
END;
```

```
FUNCTION GENASSOC X L EQFN;
LOOPIF L /= NIL
 THEN
 IF EQFN(X,FRONT(HD(L))) THEN HD(L); 1; EXIT;
 TL(L) -> L;
 CLOSE;
0;
END;

GENASSOC(% EQ %) -> ASSOC;


FUNCTION SHD X;
IF ATOM(X) THEN UNDEF;
 ELSE HD(X); CLOSE;
END;

MACRO SWAP;
MACRESULTS([;TERM1;TERM2->TERM1->TERM2;]);
END;




FUNCTION ISSTAR X;
CHARWORD(X,1) = 26;
END;



FUNCTION ISREALLINK TERM;
TOP:
IF ATOM(TERM)
 THEN
 IF TERM = NIL OR TERM = "T"
  THEN 1;
  ELSE ISINTEGER(TERM); CLOSE;
 ELSEIF HD(TERM) = "CONS"
  THEN IF ISREALLINK(HD(TL(TERM)))
   THEN HD(TL(TL(TERM))) -> TERM; GOTO TOP;
   ELSE 0; CLOSE;
 ELSE 0; CLOSE;
END;

FUNCTION ISNUMSKO X;
IF ATOM(X)
 THEN
 CHARWORD(X,1)->X;
 IF X > 40 THEN X < 47; ELSE 0; CLOSE;
 ELSE 0; CLOSE;
END;

FUNCTION ISCONS TERM;
IF ISNUMBER(TERM) OR TERM = "T"
 THEN 1;
 ELSE SHD(TERM) = "CONS" CLOSE;
END;
```

```
FUNCTION SUBST X Y Z;
VARS TEMP;
IF SUBST1(Z) THEN ELSE Z;CLOSE;
END;

FUNCTION SUBST1 Z;
IF IDENT(Y,Z)=1 THEN X; 1;
ELSEIF ATOM(Z) THEN 0;
ELSE SUBST;
LOOP1: IF Z=NIL THEN ELSE Z; TL(Z)->Z; GOTO LOOP1;CLOSE;
LOOP2: ->TEMP;
  IF TEMP=SUBST THEN 0; EXIT;
  TEMP->Z;
  IF SUBST1(HD(Z)) THEN CONS(TL(Z))->Z;GOTO LOOP3;
  ELSE GOTO LOOP2;
  CLOSE;

LOOP3:
  ->TEMP;
  IF TEMP=SUBST THEN Z;1;EXIT;
  CONS(HD(TEMP),Z)->Z;
  IF SUBST1(HD(Z)) THEN ->HD(Z); CLOSE;
  GOTO LOOP3;
CLOSE;
END;



FUNCTION APPSUB1 ALIST TERM;
VARS X;
IF ASSOCID(TERM,ALIST)
 THEN BACK(); 1;
ELSEIF ATOM(TERM)
 THEN TERM; 0;
ELSE
 APPSUB1(ALIST,HD(TERM)) -> X;
 IF LOGOR(APPSUB1(ALIST,TL(TERM)), X)
  THEN CONS(); 1;
  ELSE ERASE(); ERASE(); TERM; 0; CLOSE;
 CLOSE;
END;

FUNCTION APPSUBST;
ERASE(APPSUB1());
END;

FUNCTION INTSECTP L1 L2 TESTFN;
LOOPIF L1 /= NIL
 THEN
 IF GENMEM(HD(L1),L2,TESTFN) THEN 1; EXIT;
 TL(L1) -> L1;
 CLOSE;
0;
END;

FUNCTION UNION L1 L2 TESTFN;
LOOPIF L1 /= NIL
 THEN
 IF GENMEM(HD(L1),L2,TESTFN)
  THEN;
```

```
  ELSE HD(L1)::L2->L2;CLOSE;
 TL(L1)->L1;
 CLOSE;
L2;
END;


FUNCTION CONSCNT L;
IF ATOM(L)
 THEN 0;
 ELSE 1+CONSCNT(HD(L))+CONSCNT(TL(L)); CLOSE;
END;


FUNCTION PRSEQUEN STR LIST PRFN;
IF VERBOSE
 THEN
 POPTTON();
 NL(4);
 PRSTRING(STR);
LOOP:
 PRFN(HD(LIST));
 TL(LIST) -> LIST;
 IF LIST = NIL THEN PRSTRING('.');NL(2);EXIT;
 IF TL(LIST) = NIL
  THEN PRSTRING(' AND ');
  ELSE PRSTRING(', ');
  CLOSE;
 GOTO LOOP;
 CLOSE;
END;


FUNCTION LISPPRIM TERM;
IF ATOM(TERM)
 THEN IF TERM = NIL OR TERM = "T" THEN 1; ELSE ISNUMBER(TERM);CLOSE;
 ELSE
 HD(TERM) -> TERM;
 IF TERM = "CAR" OR TERM = "CDR" OR TERM = "CONS" OR
    TERM = "EQUAL" OR TERM = "COND"
   THEN 1;
   ELSE 0; CLOSE;
  CLOSE;
END;


MACRO PPRDEF;
PPR(PROP("DEFN",ITEMREAD()));
END;
```

```
[/ TYPE]  TRACK  9                                        [ 21.17 14 SEPT 1973]
CREATED   24.13   31  5 1973


COMMENT 'THIS FILE CONTAINS THE FUNCTIONS WHICH DECIDE IF AN EXPRESSION
IS BOOLEAN, NUMERIC, OR OF SOME OTHER TYPE.  THE FUNCTION "TYPEEXPR"
ACTUALLY WRITES LISP FUNCTIONS.`;


VARS EVALUATE NORMALIZE REDUCE CONSFN ATOMFN PROPNAME NUMERIC BOOLEAN;



COMMENT 'THIS IS A GENERAL FUNCTION FOR DECIDING IF AN EXPRESSION IS
BOOLEAN OR NUMERIC.  THESE ARE SUCH COMMON TYPES IT WAS DECIDED TO
CHECK FOR THEM EXPLICITLY.  IT IS JUST A SPECIALIZATION OF THE GENERAL
TYPE FUNCTION.  ESSENTIALLY IT JUST CHECKS THAT EVERY POSSIBLE OUTPUT
FROM THE EXPRESSION SATISFIES THE APPROPRIATE PROPERTY.`;
FUNCTION GENTYPR1 TERM;
VARS FUNSYM;
IF ATOM(TERM) THEN ATOMFN(TERM); EXIT;
HD(TERM) -> FUNSYM;
IF FUNSYM = "CONS" THEN CONSFN(TERM);
ELSEIF FUNSYM = "CAR" OR FUNSYM = "CDR" THEN 0;
ELSEIF FUNSYM = "EQUAL" THEN 1;
ELSEIF FUNSYM = "COND"
 THEN
 IF GENTYPR1(HD(TL(TL(TERM))))
  THEN GENTYPR1(HD(TL(TL(TL(TERM)))));
  ELSE 0; CLOSE;
ELSE
PROP(PROPNAME,FUNSYM) -> FOO1;
IF FOO1 /= UNDEF THEN FOO1; EXIT;
PROP("DEFN",FUNSYM) -> FOO1;
IF FOO1 = UNDEF THEN 0 -> PROP(PROPNAME,FUNSYM); 0; EXIT;
1 -> PROP(PROPNAME,FUNSYM);
IF GENTYPR1(HD(TL(TL(FOO1))))
 THEN 1;
 ELSE 0 -> PROP(PROPNAME,FUNSYM); 0; CLOSE;
CLOSE;
END;

FUNCTION GENTYPER TERM ATOMFN CONSFN PROPNAME;
GENTYPR1(TERM);
END;

GENTYPER(% LAMBDA TERM; IF TERM = NIL OR TERM = 0 OR TERM = 1 OR TERM = "T"
            THEN 1; ELSE 0; CLOSE; END,
           LAMBDA TERM; IDENT(TERM,1) = 1; END,
           "BOOLEAN" %) -> BOOLEAN;

GENTYPER(% LAMBDA TERM; IF ISNUMBER(TERM) OR ISNUMSKO(TERM) OR TERM = "T"
            THEN 1;ELSE TERM = NIL; CLOSE; END,
           LAMBDA TERM; IF IDENT(HD(TL(TERM)),NIL) = 1
            THEN NUMERIC(HD(TL(TL(TERM)))); ELSE 0; CLOSE;END,
           "NUMERIC" %) -> NUMERIC;
```

```
COMMENT 'THE FUNCTIONS "BOOLEAN" AND "NUMERIC" (ABOVE) ARE JUST
INSTANCES OF THE MORE GENERAL GENTYPER.';

COMMENT 'THE FUNCTION NORMALATE JUST EVALS, NORMALIZES AND REDUCES AN
EXPRESSION TO DEATH.';
FUNCTION NORMALATE TERM;
VARS L;
LOOP:
TERM -> L;
REDUCE(NORMALIZE(EVALUATE(TERM))) -> TERM;
IF EQUAL(TERM,L) THEN L; EXIT;
GOTO LOOP;
END;


COMMENT 'THIS IS THE WORKHORSE OF THE FUNCTION WHICH WRITES NEW FUNCTIONS.
FOR EVERY OUTPUT OF AN EXPRESSION, TYPEEXP1 PRODUCES A PIECE OF
CODE WHICH RECOGNIZES THAT OUTPUT.  THE VARIABLE "X" IS USED
TO REPRESENT THE STRUCTURE BEING INSPECTED.  IT WILL BECOME THE
LOCAL VARIABLE OF THE RECURSIVE FUNCTION PRODUCED.  NOTE THAT WHEN
THE FUNCTION ENCOUNTERS A NON-PRIM FUNCTION WHICH HAS NOT YET
BEEN TYPED IT GIVES THE NEW FUNCTION A TYPE FUNCTION (ON THE PROPERTY
LIST) AND THEN WRITES THE DEFINITION OF THAT FUNCTION.  THUS, RECURSIVE
CALLS OF THE FUNCTION BEING TYPED ARE IDENTIFIED AS ALREADY HAVING A
TYPE FUNCTION -- NAMELY, THE ONE BEING WRITTEN.  NOTE THAT AFTER
THE FUNCTION BODY HAS BEEN WRITTEN NORMALATE IS USED TO OPTIMIZE THE CODE.';
FUNCTION TYPEEXP1 TERM;
VARS TYPENAME TYPEDEFN FUNSYM PROVEFNS DEFN;
IF ATOM(TERM)
 THEN
 IF ISNUMBER(TERM) OR TERM = NIL OR TERM = "T"
  THEN [% "EQUAL", "X", TERM %];
  ELSE "T"; CLOSE;
 EXIT;
HD(TERM) -> FUNSYM;
IF FUNSYM = "CAR" OR FUNSYM = "CDR"
 THEN "T";
ELSEIF FUNSYM = "CONS"
 THEN
 IF NUMERIC(TERM)
  THEN [% "EQUAL", "X", TERM %];
  ELSE [% "COND", "X", [% "COND", SUBST([CAR X],"X",
    TYPEEXP1(HD(TL(TERM)))), SUBST([CDR X],"X",
    TYPEEXP1(HD(TL(TL(TERM))))), NIL %], NIL %]; CLOSE;
ELSEIF FUNSYM = "EQUAL"
 THEN [COND X [EQUAL X T] T];
ELSEIF FUNSYM = "COND"
 THEN
 [% "COND", TYPEEXP1(HD(TL(TL(TL(TERM))))), "T",
          TYPEEXP1(HD(TL(TL(TERM)))) %];
ELSEIF BOOLEAN(TERM)
 THEN [BOOLEAN X];
ELSEIF NUMERIC(TERM)
 THEN [NUMBERP X];
ELSE
PROP("TYPEFN",FUNSYM) -> TYPENAME;
IF TYPENAME /= UNDEF THEN TYPENAME :: [X]; EXIT;
PROP("DEFN",FUNSYM) -> DEFN;
```

```
IF DEFN = UNDEF
 THEN "CONSTTRUE" -> PROP("TYPEFN",FUNSYM); "T"; EXIT;
GENSYM(FUNSYM,"TYPE") -> TYPENAME;
TYPENAME -> PROP("TYPEFN",FUNSYM);
1 -> PROP("BOOLEAN",TYPENAME);
NORMALATE(SUBST(NIL,[%TYPENAME, "X"%],TYPEEXP1(HD(TL(TL(DEFN))))))
  -> TYPEDEFN;
IF IDENT(TYPEDEFN,"T") = 1 OR EQUAL(TYPEDEFN,[CONSTTRUE X])
 THEN
 "CONSTTRUE" -> PROP("TYPEFN",FUNSYM);
 "T";
 ELSE
 DEFINE(TYPENAME::("LAMBDA"::([X]::(TYPEDEFN::NIL))::NIL));
 TYPENAME::[X];
 CLOSE;
CLOSE;
END;



COMMENT 'THIS IS THE TOP-LEVEL FUNCTION FOR TYPING.  IT LETS TYPEEXP1
DO THE WORK AND FILTERS OUT THE CONSTANT TRUE FUNCTION';
FUNCTION TYPEEXPR TERM;
TYPEEXP1(TERM) -> TERM;
TERM;
IF NOT(ATOM(TERM)) AND LISPPRIM(TERM)
 THEN
 NORMALATE();
 CLOSE;
-> TERM;
IF IDENT(TERM,"T") = 1
 THEN [CONSTTRUE X];
 ELSE TERM; CLOSE;
END;
```

[/ SORTDEFS] TRACK 9 [21.17 14 SEPT 1973]
CREATED 15.36 8 5 1973

```
VARS OLDMARG2 OLDDEFINE OLDPPRSPCHAR;
COMPILE(LIBRARY([ALLSORT]));
DEFINE -> OLDDEFINE;
IDENTFN -> DEFINE;
PPRSPCHAR -> OLDPPRSPCHAR;
MARG2 -> OLDMARG2;
79 -> MARG2;
16 -> PPRSPCHAR;
DTRACK(36);
DOUT([/DEFS]) -> DDF2;
DDF2 -> CUCHAROUT;
APPLIST(ALLSORT([% COMPILE(DIN([/DEFS])) %],
                  LAMBDA X Y; ALFER(HD(X),HD(Y));END),
        LAMBDA X;
        NL(2);PRSTRING('DEFINE`);NL(1);PRSTRING('(`);
        PPRIND(X,1,2);PRSTRING(');`);
        END);
DDF2(TERMIN);
OLDMARG2 -> MARG2;
OLDPPRSPCHAR -> PPRSPCHAR;
CHAROUT -> CUCHAROUT;
OLDDEFINE -> DEFINE;
```

```
COMMENT 'THIS FILE CREATES THE GENSYM FUNCTION.  THE FUNCTION IS USED
TO GENERATE NEW ATOMS FOR SKOLEM CONSTANTS AND FUNCTION
NAMES.  THE SECOND ARGUMENT IS USUALLY 0 MEANING GENERATE THE
NEXT ATOM STARTING WITH THE TOPWORD.  IF THE SECOND ARGUMENT IS
NOT A NUMBER, THE TWO WORDS ARE CONCATENATED TO FORM THE NEW
SYMBOL.';

VARS GENALIST GLBGENALIST;

NIL -> GLBGENALIST;
NIL -> GENALIST;

FUNCTION NOCHARS X;
VARS R;
X//10->X->R;
IF X THEN NOCHARS(X)->X;R;X+1;
 ELSE R;1; CLOSE;
END;

FUNCTION GENSYM TOPWORD BTMWORD;
VARS CNT;
IF BTMWORD = 0
 THEN
 IF ASSOC(TOPWORD,GENALIST)
 THEN
 -> CNT;
 BACK(CNT) + 1 -> BTMWORD;
 BTMWORD -> BACK(CNT);
 ELSE
 CONSPAIR(TOPWORD,1) :: GENALIST -> GENALIST;
 1 -> BTMWORD;
 CLOSE;
 CLOSE;
IF ISNUMBER(BTMWORD)
 THEN
 CONSWORD(NOCHARS(BTMWORD)) -> BTMWORD;
 CLOSE;
IF DATALENGTH(TOPWORD) + DATALENGTH(BTMWORD) > 8
 THEN
 DESTWORD(TOPWORD) -> CNT;
 LOOPIF CNT > 4 THEN CNT-1->CNT;ERASE();CLOSE;
 DESTWORD(BTMWORD)+CNT -> CNT;
 LOOPIF CNT > 8 THEN CNT-1->CNT;ERASE();CLOSE;
 ELSE
 DESTWORD(TOPWORD)->CNT;
 DESTWORD(BTMWORD)+CNT->CNT;
 CLOSE;
CONSWORD(CNT);
END;

FUNCTION GLBGENSYM;
VARS GENALIST;
GLBGENALIST -> GENALIST;
GENSYM();
```

```
GENALIST->GLBGENALIST;
END;


GENSYM -> GENSKO;
```

[/ PROPS] TRACK 9 [21.18 14 SEPT 1973]
CREATED 9.33 8 5 1973

```
COMMENT 'THIS FILE IMPLEMENTS PROPERTY LISTS IN POP-2 USING MEANING';

FUNCTION PROP PROPNAME WORD;
VARS X;
MEANING(WORD) -> X;
IF X = UNDEF THEN NIL -> X; X-> MEANING(WORD);CLOSE;
LOOPIF X /= NIL
 THEN IF HD(X) = PROPNAME THEN HD(TL(X)); EXIT;
 TL(TL(X))->X;
 CLOSE;
PROPNAME :: (UNDEF :: MEANING(WORD)) -> MEANING(WORD);
UNDEF;
END;

LAMBDA VAL PROPNAME WORD;
VARS X;
MEANING(WORD) -> X;
IF X = UNDEF THEN NIL->X;X->MEANING(WORD);CLOSE;
LOOPIF X /= NIL
 THEN
 IF HD(X) = PROPNAME THEN VAL -> HD(TL(X));EXIT;
 TL(TL(X))->X;
 CLOSE;
PROPNAME :: (VAL :: MEANING(WORD))->MEANING(WORD);
END;->UPDATER(PROP);
```

```
[PPR]  TRACK  9                                        [ 21.18 14 SEPT 1973]
CREATED   18.53    7  5 1973


COMMENT 'THIS IS THE WORLDS BEST PRETTY PRINT ROUTINE.  IT PRINTS OUT
LISP EXPRESSIONS VERY NEATLY AND VERY FAST.  DO NOT BOTHER TO GET
BOGGED DOWN IN IT UNLESS YOU WANT TO KNOW ALL THERE IS TO KNOW ABOUT
PRETTY PRINTING.  SEE BOB BOYERS MEMO ON IT FROM DCL.  FOR THE
THEOREM PROVERS PURPOSES IT IS SUFFICIENT TO KNOW THAT PPR
PRINTS A LISP EXPRESSION.';


VARS NILCONS X TEMP1 PPRPACK PPRDL ENDLIST STARTLIS ADDLINES REMAINDE FLATSIZE
RPARCNT SPACELEF GRECCNT PPRMAX1 PPRMAXLNS MARG2 PPRSTRIP
PPRSPCHAR PPRSP STARTLIST NEXTIND NEXTNODE PPRATOM PPRJUMP PPRLINES
PPR PPRFLAG;
NIL :: NIL ->NILCONS;

30 -> PPRSPCHAR;
16->PPRMAXLNS;
60->MARG2;
INITC(80)->PPRSTRIP;



FUNCTION PPR1 FMLA RPARCNT;
VARS NODENAME DLHDFMLA RUNFLAT MINREM L RUNSTART RUNEND;
GRECCNT->NODENAME;
GRECCNT+1->GRECCNT;
IF ATOM(HD(FMLA)) THEN PPRDL(HD(FMLA))+1->DLHDFMLA;
ELSE 0->DLHDFMLA;
   FMLA->TL(NILCONS);
   NILCONS->FMLA;
CLOSE;
IF TL(FMLA)=NIL THEN RPARCNT+DLHDFMLA->FLATSIZE;
   SPACELEFT-FLATSIZE->REMAINDER;EXIT;
DLHDFMLA->RUNFLAT;
SPACELEFT-DLHDFMLA->MINREM;
SPACELEFT-1->SPACELEFT;
FMLA->L;
LOOPFLAT:
TL(L)->L;
IF L=NIL THEN SPACELEFT+1->SPACELEFT;
   IF RUNFLAT =< SPACELEFT THEN                AND RUNFLAT =< FORCEIN
      RUNFLAT->FLATSIZE;
      SPACELEFT-RUNFLAT->REMAINDER;
   ELSE PPRPACK()::NIL->STARTLIST;
      STARTLIST->ENDLIST;
      FALSE->FLATSIZE;
   CLOSE;
EXIT;
IF ATOM(HD(L)) THEN
   PPRDL(HD(L))->TEMP1; TEMP1+1+RUNFLAT->RUNFLAT;
```

```
   SPACELEFT-TEMP1->TEMP1;
   IF TL(L)=NIL THEN RPARCNT+RUNFLAT->RUNFLAT;
      TEMP1-RPARCNT->TEMP1;
   CLOSE;
   IF TEMP1<MINREM THEN TEMP1->MINREM;CLOSE;
   GOTO LOOPFLAT;
ELSE PPR1(HD(L),IF TL(L)=NIL THEN RPARCNT+1;ELSE 1;CLOSE);
IF REMAINDER<MINREM THEN REMAINDER->MINREM;CLOSE;
   IF FLATSIZE THEN FLATSIZE+1+RUNFLAT->RUNFLAT;
      GOTO LOOPFLAT;
   CLOSE;
CLOSE;
STARTLIST->RUNSTART;
ENDLIST->RUNEND;
LOOPIND:
TL(L)->L;
IF L=NIL THEN
   PPRPACK()::RUNSTART->STARTLIST;
   RUNEND->ENDLIST;
   FALSE->FLATSIZE;
   SPACELEFT+1->SPACELEFT;
EXIT;
IF ATOM(HD(L)) THEN SPACELEFT-PPRDL(HD(L))->TEMP1;
   IF TL(L)=NIL THEN TEMP1-RPARCNT->TEMP1;CLOSE;
   IF TEMP1<MINREM THEN TEMP1->MINREM;CLOSE;
   GOTO LOOPIND;
CLOSE;
PPR1(HD(L),IF TL(L)=NIL THEN RPARCNT+1;ELSE 1;CLOSE);
IF REMAINDER<MINREM THEN REMAINDER->MINREM;CLOSE;
IF FLATSIZE THEN
ELSE STARTLIST->TL(RUNEND);ENDLIST->RUNEND;
CLOSE;
GOTO LOOPIND;
END;
```

```
FUNCTION PPRPACK;
LOGOR(LOGSHIFT(IF MINREM<DLHDFMLA THEN MINREM+1;0->REMAINDER;
         ADDLINES(LENGTH(FMLA)-1);
      ELSE 17+DLHDFMLA; MINREM-DLHDFMLA->REMAINDER;
         ADDLINES(LENGTH(FMLA)-2);
      CLOSE,13),NODENAME)
END;
```

```
FUNCTION PPR2 FMLA MARG1;
VARS NONLFLAG INDFLAG PROGFLAG;
IF ATOM(FMLA) THEN PPRATOM(FMLA);EXIT;
IF HD(FMLA)="PROG" THEN MARG1; ELSE NIL;CLOSE->PROGFLAG;
IF GRECCNT=NEXTNODE THEN
   LOGAND(NEXTIND,15)+MARG1->MARG1;
   1->INDFLAG;
```

LBIN DISC DUMP 9
DAYTIME DAY MONTH YEAR
DUMPIN (9, [9 DAYTIME ...])
NO NO

```
   LOGAND(NEXTIND,16)->NONLFLAG;
   TL(STARTLIST)->STARTLIST;
   IF NIL = STARTLIST THEN
   ELSE LOGAND(HD(STARTLIST),2:1111111111111)->NEXTNODE;
        LOGSHIFT(HD(STARTLIST),-13)->NEXTIND;
   CLOSE;
ELSE 0->INDFLAG;1->NONLFLAG;
CLOSE;
GRECCNT+1->GRECCNT;
CUCHAROUT(59);
IF ATOM(HD(FMLA)) THEN
   PPRATOM(HD(FMLA));
   TL(FMLA)->FMLA;
   IF FMLA=NIL THEN CUCHAROUT(61);EXIT;
   IF NONLFLAG THEN CUCHAROUT(16);
   ELSE CUCHAROUT(17);PPRSP(MARG1);1->SUBSCRC(MARG1+1,PPRSTRIP);
   CLOSE;
CLOSE;
LOOP:
IF NONLFLAG THEN ELSEIF TL(FMLA)=NIL THEN 0->SUBSCRC(MARG1+1,PPRSTRIP);CLOSE;
PPR2(HD(FMLA),MARG1);
TL(FMLA)->FMLA;
IF FMLA=NIL THEN CUCHAROUT(61);
EXIT;
IF INDFLAG THEN CUCHAROUT(17);
   PPRSP(IF PROGFLAG/=NIL AND ATOM(HD(FMLA)) THEN PROGFLAG;
      ELSE MARG1; CLOSE;);
ELSE CUCHAROUT(16);CLOSE;
GOTO LOOP;
END;




FUNCTION ADDLINES CNT;
CNT+PPRLINES->PPRLINES;
IF PPRLINES>PPRMAX1 THEN PPRJUMP();CLOSE;
END;




FUNCTION PPRIND FMLA MARG1 RPARCNT;
VARS X;
IF ATOM(FMLA)THEN PR(FMLA);EXIT;
IF HD(FMLA)="COND" THEN PPRMAXLNS ELSE 1000000;CLOSE
->PPRMAX1;
JUMPOUT(LAMBDA;PRSTRING('(TOO BIG)');END,0)->PPRJUMP;
0->PPRLINES;
0->GRECCNT;
MARG2-MARG1->SPACELEFT;
PPR1(FMLA,RPARCNT+1);
IF FLATSIZE THEN PR(FMLA);EXIT;
FORALL X 1 1 80; 0->SUBSCRC(X,PPRSTRIP);CLOSE;
LOGAND(HD(STARTLIST),2:1111111111111)->NEXTNODE;
LOGSHIFT(HD(STARTLIST),-13)->NEXTIND;
```

```
0->GRECCNT;
PPR2(FMLA,MARG1);
END;



PPRIND(% 0, 0 %)->PPR;



FUNCTION PPRDL L;
VARS CNT CUCHAROUT;
IF ISNUMBER(L) THEN -1->CNT;
LAMBDA X;CNT+1->CNT;END->CUCHAROUT;
PR(L);
CNT;
ELSE DATALENGTH(L);CLOSE;
END;



FUNCTION PPRATOM L;
VARS CUCHAROUT OCUCHAROUT;
IF ISNUMBER(L) THEN
   CUCHAROUT->OCUCHAROUT;
   LAMBDA X; IF X=16 THEN ELSE OCUCHAROUT(X);CLOSE;END
   ->CUCHAROUT;
ELSEIF DATAWORD(L)="CSTRIP" THEN PRSTRING(L);
EXIT;
PR(L);
END;



FUNCTION PPRSP N;
0->PPRFLAG;
FORALL X 1 1 N;
IF SUBSCRC(X,PPRSTRIP) AND NOT(PPRFLAG) THEN
   CUCHAROUT(PPRSPCHAR);1->PPRFLAG;
ELSE CUCHAROUT(16);0->PPRFLAG;
CLOSE;
CLOSE;
END;
```

```
[/ IDENT]  TRACK  9                                        [ 21.18 14 SEPT 1973]
CREATED   19.47   17  4 1973



COMMENT '"IDENT" IS A FAIRLY IMPORTANT FUNCTION.  IT MERELY RECOGNIZES
WHEN TWO TERMS ARE IDENTICAL (AND THEREFORE EQUAL), WHEN THEY
CANNOT POSSIBLY BE EQUAL (E.G., A CONS VERSUS A NIL), OR OF
UNKNOWN RELATIONSHIP SYNTACTICALLY.  IT RETURNS 1 IF THEY ARE
IDENTICAL (IT KNOWS ABOUT INTEGERS BEING CONSES, ETC), 0 IF
THEY ARE OF UNKNOWN RELATIONSHIP, AND NIL IF THEY ARE DEFINATELY
UNEQUAL.';

VARS MEMBERID ASSOCID;

FUNCTION OCCUR CONST TERM;
IF IDENT(CONST,TERM)=1 THEN 1;EXIT;
IF ATOM(TERM) THEN 0; EXIT;

LOOPIF (TL(TERM)->TERM; TERM /= NIL)
 THEN
 IF OCCUR(CONST,HD(TERM)) THEN 1; EXIT;
 CLOSE;
0;
END;

FUNCTION OCCURCONS TERM1 TERM2;
IF SHD(TERM2) /= "CONS"
 THEN IDENT(TERM1,TERM2) = 1
ELSEIF OCCURCONS(TERM1,HD(TL(TERM2)))
 THEN 1;
 ELSE OCCURCONS(TERM1,HD(TL(TL(TERM2)))); CLOSE;
END;



FUNCTION IDENT TERM1 TERM2;
VARS FUNSYM;
TOP:
IF TERM1 = TERM2 THEN 1; EXIT;
IF ATOM(TERM1)
 THEN
 IF TERM1 = NIL THEN 0 -> TERM1;
 ELSEIF TERM1 = "T" THEN 1 -> TERM1; CLOSE;
 IF ATOM(TERM2)
  THEN
  IF TERM2 = NIL THEN 0 -> TERM2;
  ELSEIF TERM2 = "T" THEN 1 -> TERM2; CLOSE;
  IF EQ(TERM1,TERM2) THEN 1;
  ELSEIF ISNUMBER(TERM1)
   THEN IF ISNUMBER(TERM2)
    THEN NIL;
    ELSE 0; CLOSE;
   ELSE 0; CLOSE;
 ELSEIF (L1: HD(TERM2) = "CONS")
  THEN
  IF TERM1
   THEN
```

```
    IF ISNUMBER(TERM1)
     THEN
     IDENT(0,HD(TL(TERM2))) -> FOO1;
     IF EQ(FOO1,1)
      THEN
           TERM1 - 1 -> TERM1;
      HD(TL(TL(TERM2))) -> TERM2;
      GOTO TOP;
      ELSE FOO1; CLOSE;
     ELSEIF OCCURCONS(TERM1,TERM2) THEN NIL; ELSE 0; CLOSE;
    ELSE NIL; CLOSE;
   ELSE 0; CLOSE;
ELSEIF ATOM(TERM2)
 THEN
 IF TERM2 = NIL THEN 0 -> TERM2;
 ELSEIF TERM2 = "T" THEN 1 -> TERM2; CLOSE;
 SWAP;
 GOTO L1;
ELSEIF HD(TERM1) = HD(TERM2)
 THEN
 HD(TERM1) -> FUNSYM;
 1;
 LOOPIF (TL(TERM1) -> TERM1; TERM1 /= NIL)
  THEN
  TL(TERM2) -> TERM2;
  IDENT(HD(TERM1),HD(TERM2)) -> FOO1;
  IF FOO1 /= 1
  THEN
  IF FUNSYM = "CONS"
   THEN
   IF FOO1 = NIL THEN ERASE(); NIL; EXIT;
   LOGAND(FOO1);
   ELSE ERASE(); 0; EXIT;
  CLOSE;
  CLOSE;
ELSEIF HD(TERM2) = "CONS" OR HD(TERM1) = "CONS" AND (SWAP;1;)
 THEN
 IF OCCURCONS(TERM1,TERM2)
  THEN NIL;
  ELSE 0; CLOSE;
ELSE 0; CLOSE;
END;


COMMENT 'THIS IS JUST AN EQUALITY LIKE OPERATION WHICH IS TRUE IF ITS
TWO ARGUMENTS ARE IDENT AND FALSE OTHERWISE.';
OPERATION 7 ==;
IDENT() = 1;
END;



GENASSOC(% NONOP == %) -> ASSOCID;
GENMEM(% NONOP == %) -> MEMBERID;
```

```
[ALAN]  TRACK  9                                        [ 21.19 14 SEPT 1973]
CREATED   16.28   16  2 1973



DEFINE([SLESS [LAMBDA [X Y] [NOT [LTE Y X]]]]);

DEFINE([SUC [LAMBDA [X] [CONS NIL X]]]);

DEFINE([PRE [LAMBDA [X] [COND X [CDR X] NIL]]]);

DEFINE([DIFF [LAMBDA [X Y] [CDRN Y X]]]);

DEFINE([EQN [LAMBDA [X Y] [EQUAL [LENGTH X] [LENGTH Y]]]]);

DTRACK(256);
INCHARITEM(DIN([DATA]))->DDF1;
POPVAL(FNTOLIST(LAMBDA;VARS X1;
DDF1()->X1;
IF X1 = "EQ" THEN "EQN";
ELSEIF X1 = "LESS" THEN "LTE";
ELSEIF X1 = "ADD" THEN "APPEND";
ELSEIF X1 = "IFF" THEN "EQUAL";
ELSEIF X1 = "FUNCTION" THEN "GOON";
ELSE X1; CLOSE;
END));

DTRACK(9);
```