

16.18HRS. 2 NOV 1973.

** RSB **

[FILE DUMP]

[/ PROPS]

[/ GENSYM]

[/ MACCONS]

[/ APPFILE]

[/ INPUT]

[/ GEN]

[PPR]

[/ TYPE]

[/ METAGEN]

[/ EVAL]

[/ PROPS] TRACK 9
CREATED 14.21 1 11 1973.

[16.18 2 NOV 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->MEANING(WORD);UNDEF;EXIT;  
  LOOPIF X/=NIL  
    THEN IF HD(X)=PROPNAME THEN HD(TL(X));EXIT;  
    TL(TL(X))->X;  
  CLOSE;  
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);
```

[/ GENSYM] TRACK 9
CREATED 14.20 1 11 1973

[16.18 2 NOV 1973]

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;

[/ MACCONS] TRACK 9
CREATED 14.20 1 11 1973

[16.18 2 NOV 1973]

```
FUNCTION MACCONS X Y;
VARS HDY;
IF ATOM(X) THEN
  SHD(Y)->HDY;
  IF X="QUOTE" THEN
    IF ATOM(HDY) THEN
      IF ISVALUE(HDY) THEN HDY;
      ELSE CONS(X,Y);CLOSE;
    ELSE [%"CONS",MACCONS("QUOTE",HD(HDY)::NIL),MACCONS("QUOTE"
      ,TL(HDY)::NIL)%];
    CLOSE;
  ELSEIF X="COND" THEN
    IF Y=NIL THEN NIL;
    ELSE
      IF LENGTH(HDY)>2 THEN
        [%HD(HDY),"PROGN"::TL(HDY)%]->HDY;CLOSE;
      IF HD(HDY)=T THEN HD(TL(HDY));
      ELSE [%"IF",HD(HDY),HD(TL(HDY)),MACCONS("COND",TL(Y))%];
      CLOSE;
    CLOSE;

  ELSEIF X="AND" THEN
    IF TL(Y)=NIL THEN HDY;
    ELSE [%"IF",HDY,MACCONS("AND",TL(Y)),NIL%];CLOSE;
  ELSEIF X="OR" THEN
    IF TL(Y)=NIL THEN HDY;
    ELSE [%"IF",HDY,T,MACCONS("OR",TL(Y))%];
    CLOSE;
  ELSEIF X="LIST" THEN
    IF Y=NIL THEN NIL;
    ELSE [%"CONS",HDY,MACCONS("LIST",TL(Y))%];
    CLOSE;
  ELSE CONS(X,Y);CLOSE;
ELSEIF EQUAL(X,[QUOTE -]) AND ISNUMBER(HDY) THEN
  CONS(0-HDY,TL(Y));
ELSE CONS(X,Y);CLOSE;
END;
```

```
FUNCTION MACEX FORM;
IF ATOM(FORM) THEN FORM;
ELSEIF HD(FORM)=""" THEN
  MACCONS(MACCONS("QUOTE",HD(TL(FORM))::NIL),MACEX(TL(TL(FORM))));
ELSEIF HD(FORM)="-" THEN
  MACCONS(0-HD(TL(FORM)),MACEX(TL(TL(FORM))));
ELSEIF HD(FORM)="QUOTE" THEN MACCONS(HD(FORM),TL(FORM));
ELSE MACCONS(MACEX(HD(FORM)),MACEX(TL(FORM)));
CLOSE;
END;
```

[/ APPFILE] TRACK 9
CREATED 14.19 1 11 1973

[16.18 2 NOV 1973]

```
FUNCTION READ;  
VARS X;  
LOOP:  
ITEMREAD()->X;  
IF X="]" THEN GOTO LOOP;  
ELSEIF X="[" THEN X::PROGLIST->PROGLIST;  
LISTREAD();  
ELSE X;CLOSE;  
END;
```

```
FUNCTION APPFILE FILEN APPFILEF;  
VARS PROGLIST A B C;  
O->XAPPFILE;  
DISC(FILEN)->DDF1;  
FNTOLIST(INCHARITEM(DDF1))->PROGLIST;  
LOOPIF NOT(NULL(PROGLIST)) AND NOT(XAPPFILE) THEN APPFILEF(READ());CLOSE;  
END;
```

[/ INPUT] TRACK 9
CREATED 14.18 1 11 1973

[16.18 2 NOV 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 SAMENAME NAME1 NAME2;
LOOP:
IF NAME1=NIL THEN 1;EXIT;
IF NAME2=NIL THEN 0;EXIT;
IF HD(NAME1)/=HD(NAME2) THEN 0;EXIT;
TL(NAME1)->NAME1;
TL(NAME2)->NAME2;
GOTO LOOP;
END;

FUNCTION GETTHM THMNAME;
VARs PROGLIST TRK;
DISCUSER->TRK;
DTRACK(THMTRACK);
IF HD(THMNAME)="ALL"
THEN [%APPFILE(THMFILE,IDENTFN)%];DTRACK(TRK);
->TRK;
IF TL(THMNAME)/=NIL
THEN
HD(TL(THMNAME))->THMNAME;
LOOPIF NOT(SAMENAME(THMNAME,HD(HD(TRK)))) THEN TL(TRK)->TRK;CLOSE;
CLOSE;
TRK;
EXIT;
APPFILE(THMFILE,LAMBDA X;
IF SAMENAME(THMNAME,HD(X)) THEN 1->XAPPFILE;X->THMNAME;CLOSE;
END);
DTRACK(TRK);
IF XAPPFILE=1 THEN HD(TL(THMNAME));EXIT;
ERRFUN(THMNAME,57);
END;

FUNCTION ADDEFS;

DCOMP([ADDEFS]);
END;

[/ GEN] TRACK 9
CREATED 14.17 1 11 1973

[16.19 2 NOV 1973]

COMMENT 'THESE ARE GENERAL PURPOSE LIST PROCESSING FUNCTIONS.
EXCEPT FOR OCCUR, SUBST, AND APPSUBST, THEY HAVE NO
SPECIAL KNOWLEDGE OF THE LISP SUBSET USED. THE EXCEPTIONS
KNOW OF THE EXISTENCE OF "QUOTE" AND HOW IT WORKS IN THE LISP.';

```
FUNCTION NCONS LIST ELE;
IF LIST=NIL THEN ELE::NIL;EXIT;
LIST;
LOOPIF TL(LIST)/=NIL THEN TL(LIST)->LIST;CLOSE;
ELE::NIL->TL(LIST);
END;
```

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

```
GENMEM(%NONOP=%)>MEMBER;
GENMEM(%EQUAL%)>MEMBEREQUAL;
```

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

```
IF EQUAL(CONST,TERM) THEN 1;EXIT;
IF ATOM(TERM) OR HD(TERM)="QUOTE" THEN 0;EXIT;
```

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

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

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

```
FUNCTION PAIRLIST X Y ALIST;
LOOPIF X/=NIL
  THEN
  CONSPAIR(HD(X),HD(Y))::ALIST->ALIST;
  TL(X)->X;
  TL(Y)->Y;
  CLOSE;
ALIST;
END;
```

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

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

```
GENASSOC(%NONOP=%)>ASSOC;
GENASSOC(%EQUAL%)>ASSOCEQUAL;
```

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

```
FUNCTION SUBST1 Z;
IF EQUAL(Y,Z) THEN X;1;
ELSEIF ATOM(Z) OR HD(Z)="QUOTE" 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) OR HD(TERM)="QUOTE"
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;

```

[PPR] TRACK 9
CREATED 14.16 1 11 1973

[16.19 2 NOV 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 REMAINDER FLATSIZE
RPARCNT SPACELEFT GRECCNT PPRMAX1 PPRMAXLNS MARG2 PPRSTRIP
PPRSPCHAR PPRSP STARTLIST NEXTIND NEXTNODE PPRATOM PPRJUMP PPRLINES
PPR PPRFLAG FORCEIN;
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
IF HD(FMLA)="LAMBDA" THEN 0;
ELSE PPRDL(HD(FMLA))+1;CLOSE;->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 AND RUNFLAT=<FORCEIN THEN
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;
LOGAND(NEXTIND,16)->NONLFLAG;
TL(STARTLIST)->STARTLIST;
IF NIL=STARTLIST THEN
  ELSE LOGAND(HD(STARTLIST),2:11111111111111)->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),2:11111111111111111111)->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 PPRSTRING(L);  
  EXIT;  
PR(L);  
END;
```

```
PRSTRING->PPRSTRING;
```

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

38->FORCEIN;

VARS PROP;

OPERATION 1 PPRDEF L;

APPLIST(L,LAMBDA X;

PPR([X,PROP("DEFN",X)%1]);

NL(2);

END;

);

END;

[/ TYPE] TRACK 9
CREATED 14.15 1 11 1973

[16.2 2 NOV 1973]

COMMENT 'THIS FUNCTION RECOGNIZES BOOLEAN VALUED EXPRS. ';

```
FUNCTION BOOLEAN TERM;  
IF ATOM(TERM)  
  THEN  
    IF TERM=NIL OR TERM=0 THEN 1; ELSE TERM=T;CLOSE;  
  EXIT;  
LISPHASH(HD(TERM))SWITCH NOTBOOL NOTBOOL NOTBOOL LISPCONS BOOL  
LISPIF NONPRIM;  
  
NOTBOOL:0;RETURN;  
  
BOOL:1;RETURN;  
  
LISPCONS:TERM==[CONS NIL [CONS 20 NIL]];RETURN;  
  
LISPIF:  
IF BOOLEAN(HD(TL(TL(TERM))))  
  THEN BOOLEAN(HD(TL(TL(TL(TERM)))));  
  ELSE 0;CLOSE;  
RETURN;  
  
NONPRIM:  
PROP("BOOLEAN",HD(TERM))->FOO1;  
IF FOO1/=UNDEF THEN FOO1;EXIT;  
1->PROP("BOOLEAN",HD(TERM));  
PROP("DEFN",HD(TERM))->FOO1;  
IF FOO1/=UNDEF AND BOOLEAN(HD(TL(TL(FOO1)))) THEN 1;EXIT;  
0->PROP("BOOLEAN",HD(TERM));  
0;  
END;
```

COMMENT '(THIS FUNCTION IS NOT AS TIGHT AS IT MIGHT BE.
IN PARTICULAR, SINCE == ASSUMES ITS ARGS HAVE BEEN
EVALD, T IS NOT RECOGNIZED WHEN IT IS IN THE FORM:
[CONS NIL [CONS [CONS NIL 19] NIL]].
THIS COULD CAUSE A FUNCTION TO BE TYPED AS NON-BOOLEAN
WHEN IN FACT ITS BOOLEAN.))';

COMMENT ' "NUMERIC" RECOGNIZES NUMERIC EXPRESSIONS.
IT ASSUMES SKOLEM CONSTANTS STARTING WITH THE LETTERS
I THROUGH N ARE NUMERICALLY VALUED. IT
DOES NOT MODIFY THE PROPERTY LISTS OF NON-PRIMS
BUT DETERMINES THEIR TYPES EACH TIME, USING A LIST
(THE SECOND ARG) TO KEEP TRACK OF ITS ASSUMPTIONS. ';

COMMENT 'THE STRUCTURE OF THE SECOND ARG IS A LIST OF
NON-PRIM FN APPLICATIONS, IN WHICH THE ARGS ARE ALL
OS AND XS. IF [FOO 0 X] APPEARS IN THE LIST, THEN
FOO APPLIED TO A NUMERIC FIRST ARG AND A NON-NUMERIC
SECOND ARG, IS BEING ASSUMED TO BE NUMERIC (IE., THE
FORMALS ARE BOUND TO REPRESENTATIVE MEMBERS OF THE

TYPES OF THEIR VALUES).';

```
FUNCTION NUMSKO TERM;
IF ISWORD(TERM) AND TERM/=NIL AND TERM/=T
  THEN
    CHARWORD(TERM,1)->TERM;
    IF TERM>40 THEN TERM<47; ELSE 0;CLOSE;
    ELSE 0;CLOSE;
END;
```

```
FUNCTION NUMERIC1 TERM NUMFNLIST;
VARS X;
IF ATOM(TERM)
  THEN
    IF TERM=0 OR TERM=NIL THEN 1;
    ELSEIF TERM=T THEN 0;
    ELSEIF ISNUMBER(TERM) THEN 1;
    ELSE
      NUMSKO(TERM);
      CLOSE;
    EXIT;
```

```
LISPHASH(HD(TERM))SWITCH CAR CDR NOTNUM LISPCONS NOTNUM
LISPIF NONPRIM;
```

```
CAR:
IF NUMERIC1(HD(TL(TERM)),NUMFNLIST)
  THEN 1;
ELSEIF SHD(HD(TL(TERM)))="QUOTE"
  THEN 1;
ELSE HD(TL(TERM))=T;CLOSE;
RETURN;
```

```
CDR:
NUMERIC1(HD(TL(TERM)),NUMFNLIST);
RETURN;
```

```
NOTNUM:
0;
RETURN;
```

```
LISPCONS:
HD(TL(TERM))->FOO1;
IF FOO1=NIL OR FOO1=0 OR SHD(FOO1)="CAR"
  AND NUMERIC(HD(TL(FOO1)))
  THEN NUMERIC1(HD(TL(TL(TERM))),NUMFNLIST);
ELSE 0;CLOSE;
RETURN;
```

```
LISPIF:
IF NUMERIC1(HD(TL(TL(TERM))),NUMFNLIST)
  THEN NUMERIC1(HD(TL(TL(TL(TERM))))),NUMFNLIST)
  ELSE 0;CLOSE;
RETURN;
```

```
NONPRIM:
[%HD(TERM),APPLIST(TL(TERM),
  LAMBDA;
```

```
                IF NUMERIC1(NUMFNLIST) THEN 0; ELSE "X";CLOSE;
                END)]->TERM;
IF MEMBEREQUAL(TERM,NUMFNLIST) THEN 1;EXIT;
PROP("DEFN",HD(TERM))->X;
IF X=UNDEF THEN 0;EXIT;
APPSUBST(PAIRLIST(HD(TL(X)),TL(TERM),NIL),HD(TL(TL(X))))
->X;
NUMERIC1(X,TERM::NUMFNLIST);

END;

NUMERIC1(%NIL%)->NUMERIC;
```

COMMENT 'THIS FILE CONTAINS FUNCTIONS FOR HANDLING TERMS
WHICH REPRESENT LIST STRUCTURES IN THE LOGIC OF THE THEOREM
PROVER. AMONG THEM ARE THE POP-2 IMPLEMENTATIONS OF THE
LISP FUNCTIONS EQUAL, CAR, CDR, AND CONS.';

"T" -> T;

COMMENT 'THE FOLLOWING FUNCTION TESTS FOR SYNTACTIC IDENTITY
OF TWO TERMS--RESPECTING THE ABBREVIATIONS ALLOWED. IT RETURNS
TRUE IF AND ONLY IF THEY ARE IDENTICAL. IF IT RETURNS FALSE,
THE VARIABLE NOTIDENT CONTAINS TRUE IFF THEY ARE DECIDEDLY
NOT IDENTICAL. THE ARGS ARE ASSUMED TO HAVE BEEN EVALD.';

VARS OPERATION 7(== !=);

FUNCTION IDENT TERM1 TERM2;

IF ATOM(TERM1)

THEN

IF ATOM(TERM2)

THEN

IF TERM1=TERM2 THEN GOTO WIN;

ELSEIF TERM1=NIL AND TERM2=0 THEN GOTO WIN;

ELSEIF TERM1=0 AND TERM2=NIL THEN GOTO WIN;

ELSEIF ISVALUE(TERM1) AND ISVALUE(TERM2) THEN GOTO LOSE;

ELSE GOTO UNKNOWN;CLOSE;

CLOSE;

SWAP;

GOTO L1;

ELSEIF ATOM(TERM2)

THEN

L1:

IF EXPLCONS(TERM1)

THEN

IF TERM2=NIL OR TERM2=0 THEN GOTO LOSE;

ELSEIF HD(TERM1)="QUOTE"

THEN

IF ISVALUE(TERM2) THEN GOTO LOSE; ELSE GOTO UNKNOWN;CLOSE;

ELSEIF HD(TERM1)="CONS"

THEN GOTO OCCCHK;

ELSE GOTO UNKNOWN;CLOSE;

CLOSE;

ELSEIF HD(TERM1)=HD(TERM2)

THEN

IF HD(TERM1)="QUOTE"

THEN

IF HD(TL(TERM1))=HD(TL(TERM2))

THEN GOTO WIN;

ELSE GOTO LOSE;CLOSE;

ELSEIF HD(TERM1)="CONS"

THEN

IF IDENT(HD(TL(TERM1)),HD(TL(TERM2)))

THEN IDENT(HD(TL(TL(TERM1))),HD(TL(TL(TERM2)))));RETURN;

ELSEIF NOTIDENT

```

        THEN GOTO LOSE;
    ELSEIF (ERASE(IDENT(HD(TL(TL(TERM1))),HD(TL(TL(TERM2)))));NOTIDENT)
        THEN GOTO LOSE;
        ELSE GOTO UNKNOWN;CLOSE;
    ELSE
    LOOPIF (TL(TERM1)->TERM1;TERM1/=NIL)
        THEN
            TL(TERM2)->TERM2;
            IF IDENT(HD(TERM1),HD(TERM2)) THEN ELSE GOTO UNKNOWN;CLOSE;
            CLOSE;
        GOTO WIN;
    CLOSE;
ELSEIF HD(TERM1)="CONS" OR HD(TERM2)="CONS" AND (SWAP;1)
    THEN
        IF HD(TERM2)="QUOTE"
            THEN
                IF (ERASE(IDENT(HD(TL(TERM1)),NIL));NOTIDENT)
                    THEN GOTO LOSE;CLOSE;
                CLOSE;
            OCCCHK:
            IF OCCURCONS(TERM2,TERM1)
                THEN GOTO LOSE;
                ELSE GOTO UNKNOWN;CLOSE;
            CLOSE;

```

```
UNKNOWN:0->NOTIDENT;0;RETURN;
```

```
LOSE:1->NOTIDENT;0;RETURN;
```

```
WIN:0->NOTIDENT;1;
```

```
END;
```

```
COMMENT'THIS FUNCTION IS USED BY IDENT TO DISCOVER WHETHER
A CONS PROPERLY CONTAINS ANOTHER TERM AS A COMPONENT OF
THE CONS.';
```

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

```
IDENT->NONOP==;
```

```
LAMBDA;ERASE(IDENT());NOTIDENT;END->NONOP=/=;
```

```
GENMEM(%IDENT%)->MEMBERID;
GENASSOC(%IDENT%)->ASSOCID;
```

```
COMMENT'THIS FUNCTION RECOGNIZES EXPLICIT CONSES';
```

```
FUNCTION EXPLCONS TERM;
IF ATOM(TERM)
    THEN
```

```
IF TERM=T OR ISNUMBER(TERM) AND TERM
  THEN 1; ELSE 0;CLOSE;
ELSEIF HD(TERM)="CONS"
  THEN 1;
  ELSE HD(TERM)="QUOTE";CLOSE;
END;
```

```
COMMENT'THIS FUNCTION RECOGNIZES ATOMIC TERMS WHICH REPRESENT
SPECIFIC STRUCTURES';
```

```
FUNCTION ISVALUE TERM;
IF TERM=NIL OR TERM=T THEN 1;
  ELSE ISNUMBER(TERM);CLOSE;
END;
```

```
COMMENT'THIS FUNCTION RECOGNIZES TERMS WHICH REPRESENT
SPECIFIC OBJECTS';
```

```
FUNCTION ISSPEC TERM;
IF ATOM(TERM) THEN ISVALUE(TERM);
ELSEIF HD(TERM)="QUOTE" THEN 1;
ELSEIF HD(TERM)="CONS" AND ISSPEC(HD(TL(TERM)))
  THEN ISSPEC(HD(TL(TL(TERM)))));
  ELSE 0;CLOSE;
END;
```

```
COMMENT'THIS FUNCTION CONSTRUCTS A NEW SKOLEM CONSTANT GIVEN
THE NAME OF AN OLD ONE';
```

```
GENSYM(%0%)->GENSKO;
```

```
COMMENT'THIS FUNCTION GENERATES TERMS OF THE FORM [* N]
IN WHICH OTHER TERMS ARE HIDDEN. IT IS ASSUMED THAT TERM2 IS BOOLEAN
(BECAUSE "*" IS TREATED AS BOOLEAN).';
```

```
FUNCTION GENSTAR TERM;
STARCOUNT+1->STARCOUNT;
CONSPAIR(STARCOUNT,TERM)::STARALIST->STARALIST;
["*",STARCOUNT];
END;
```

```
1->PROP("BOOLEAN","*");
NIL->STARALIST;
0->STARCOUNT;
```

```
COMMENT'THE FOLLOWING THREE FUNCTIONS DEAL WITH THE LISP
REPRESENTATION OF WORDS. THE FIRST CONSTRUCTS THE LISP
PRINTNAME OF A (POP-2) WORD (A LIST OF INTEGERS BETWEEN 1
AND 26). THE NEXT RECOGNIZES WHEN A LIST IS A PRINTNAME.
THE LAST ONE CONVERTS A PRINTNAME TO THE POP-2
WORD.';
```

```
FUNCTION PRINTNAME I;
VARS ANS CHAR;
DESTWORD(I)->I;
NIL->ANS;
LOOPIF I
  THEN
```

```

32;.-;->CHAR;
[%"CONS",CHAR,ANS%]->ANS;
I-1->I;
CLOSE;
ANS;
END;

```

```

FUNCTION ISPRINTNAME TERM;
LOOPIF ISLINK(TERM)
  THEN
    IF HD(TERM)="CONS" AND (HD(TL(TERM))->F001;ISNUMBER(F001))
      AND F001
      THEN
        ELSE 0;EXIT;
      HD(TL(TL(TERM)))->TERM;
    CLOSE;
TERM=NIL;
END;

```

COMMENT'THE FUNCTION ABOVE ASSUMES THE LIST HAS BEEN EVALD--WHICH MEANS THAT THE ELEMENTS WILL BE POP-2 NUMBERS IF THEY REPRESENT NUMBERS, AND THE FINAL CDR WILL BE NIL RATHER THAN 0.';

```

FUNCTION MAKEWORD TERM;
VARS I;
0->I;
LOOPIF TERM/=NIL AND I/=8
  THEN
    HD(TL(TERM))+32;
    HD(TL(TL(TERM)))->TERM;
    I+1->I;
  CLOSE;
CONSWORD(I);
END;

```

COMMENT'THE FOLLOWING THREE FUNCTIONS COMPUTE THE CAR AND CDR (OF TERMS REPRESENTING EXPLICIT CONSES ONLY) AND CONS TWO TERMS TOGETHER. IT IS ASSUMED THAT THE ARGS HAVE BEEN EVALD.';

```

FUNCTION CAREFN XCONS;
IF ATOM(XCONS) OR HD(XCONS)="QUOTE" THEN NIL;
  ELSE HD(TL(XCONS));CLOSE;
END;

```

```

FUNCTION CDRFN XCONS;
IF ATOM(XCONS)
  THEN
    IF ISNUMBER(XCONS) THEN XCONS-1;
    ELSE [CONS 20 NIL];CLOSE;
  ELSEIF HD(XCONS)="QUOTE"
    THEN PRINTNAME(HD(TL(XCONS)));
    ELSE HD(TL(TL(XCONS)));CLOSE;
END;

```

```

FUNCTION CONSFN ARG1 ARG2;
IF ARG1=NIL OR ARG1=0
  THEN
    IF ISNUMBER(ARG2) THEN ARG2+1;
    ELSEIF ARG2=NIL THEN 1;

```

```

ELSEIF EQUAL(ARG2,[CONS 20 NIL]) THEN T;
ELSEIF ISPRINTNAME(ARG2) THEN [%"QUOTE",MAKEWORD(ARG2)%];
    ELSE [%"CONS",ARG1,ARG2%];CLOSE;
ELSE
IF ARG2=0 THEN NIL->ARG2;CLOSE;
[%"CONS",ARG1,ARG2%];
CLOSE;
END;

```

COMMENT'THese TWO FUNCTIONS PROVIDE QUICK RECOGNITION OF
THE LISP PRIMITIVES.');

```

FUNCTION LISPPRIM TERM;
IF ATOM(TERM)
    THEN
    IF TERM=NIL OR TERM=T THEN 1; ELSE ISNUMBER(TERM);CLOSE;
    ELSE PROP("LISPPRIM",HD(TERM))/=UNDEF;CLOSE;
END;

```

```

FUNCTION LISPHASH FNNAME;
PROP("LISPPRIM",FNNAME)->FNNAME;
IF FNNAME=UNDEF THEN PRIMCNT; ELSE FNNAME;CLOSE;
END;

```

```

1->PRIMCNT;
APPLIST([CAR CDR QUOTE CONS EQUAL IF],
    LAMBDA NAME;
    PRIMCNT->PROP("LISPPRIM",NAME);
    PRIMCNT+1->PRIMCNT;
    END);

```

COMMENT'THIS FUNCTION PUTS A FUNCTION DEFN ON THE PROP
LIST OF THE FN NAME, AFTER MACRO EXPANDING THE DEFN.');

```

FUNCTION DEFINE X;
[%"DEFN",[%"LAMBDA",HD(TL(HD(TL(X))))),
    MACEX(HD(TL(TL(HD(TL(X))))))%]%->MEANING(HD(X));
END;

```

[/ EVAL] TRACK 9
CREATED 14.13 1 11 1973

[16.21 2 NOV 1973]

NIL->PROVEFNS;

FUNCTION SYMEVAL1 TERM;

TOP:

IF ATOM(TERM)

THEN

IF ISVALUE(TERM) THEN TERM;EXIT;

IF ASSOC(TERM,ALIST) THEN BACK(); ELSE TERM;CLOSE;

RETURN;

CLOSE;

IF HD(TERM)/="IF" AND HD(TERM)/="QUOTE"

THEN

HD(TERM)::NIL;

LOOPIF (TL(TERM)->TERM;TERM/=NIL)

THEN NCONS(SYMEVAL1(HD(TERM)));CLOSE;->TERM;

CLOSE;

LISPHASH(HD(TERM))SWITCH CAR CDR QUOTE LISPCONS LISPEQUAL LISPIF

NONPRIM;

CAR:

IF HD(TL(TERM))=NIL THEN NIL;

ELSEIF NUMSKO(HD(TL(TERM))) THEN NIL;

COMMENT/NOTE THAT ONLY THE CAR OF A NUMERIC SKOLEM

CONSTANT (RATHER THAN ANY NUMERIC TERM) IS KNOWN TO BE NIL.;

ELSEIF EXPLCONS(HD(TL(TERM))) THEN CARFN(HD(TL(TERM)));

ELSE TERM;CLOSE;

RETURN;

CDR:

IF HD(TL(TERM))=NIL THEN NIL;

ELSEIF EXPLCONS(HD(TL(TERM))) THEN CDRFN(HD(TL(TERM)));

ELSE TERM;CLOSE;

RETURN;

QUOTE:

TERM;

RETURN;

LISPCONS:

CONSFN(HD(TL(TERM)),HD(TL(TL(TERM))));

RETURN;

LISPEQUAL:

IF HD(TL(TERM))=HD(TL(TL(TERM))) THEN T;

ELSEIF NOTIDENT THEN NIL;

ELSEIF EXPLCONS(HD(TL(TERM))) AND EXPLCONS(HD(TL(TL(TERM))))

THEN

APPLY([%CONSPAIR("Y1",CARFN(HD(TL(TERM))))),

```

        CONSPAIR("X1",CARFN(HD(TL(TL(TERM))))),
        CONSPAIR("Y2",CDRFN(HD(TL(TERM))))),
        CONSPAIR("X2",CDRFN(HD(TL(TL(TERM)))))]],
    LAMBDA ALIST;
    SYMEVAL1([IF [EQUAL Y1 X1] [EQUAL Y2 X2] NIL]);
    END);
ELSE TERM;CLOSE;
RETURN;

LISPIF:
SYMEVAL1(HD(TL(TERM)))->FOO1;
FOO1;
IF FOO1==NIL
    THEN ERASE();HD(TL(TL(TL(TERM))))->TERM;GOTO TOP;
ELSEIF NOTIDENT
    THEN ERASE();HD(TL(TL(TERM)))->TERM;GOTO TOP;
    ELSE
    ->FOO1;
    [%FOO1,SYMEVAL1(HD(TL(TL(TERM))))],SYMEVAL1(HD(TL(TL(TL(TERM)))))]])%]
    ->TERM;
    "IF"::TERM;
    CLOSE;
RETURN;

NONPRIM:
[%TERM,COMPLEXITY(TERM),0%]::FNSTACK->FNSTACK;
IF RECUROK()
    THEN
    PROP("DEFN",HD(TERM))->TERM;
COMMENT 'TERM NO LONGER HAS ANY RELATION TO WHAT IS BEING EVALD';
IF TERM=UNDEF THEN GOTO TOPSTACK;CLOSE;
PAIRLIST(HD(TL(TERM)),TL(HD(HD(FNSTACK))),ALIST)->ALIST;
SYMEVAL1(HD(TL(TL(TERM))))->TERM;
IF INDUCT AND TL(FNSTACK)=NIL
    THEN
    [%HD(HD(FNSTACK)),TERM%]::TOPLEXP->TOPLEXP;
    CLOSE;
APPLIST(TL(HD(HD(FNSTACK))),LAMBDA X;TL(ALIST)->ALIST;END);
IF HD(TL(TL(HD(FNSTACK)))) THEN GOTO TOPSTACK; ELSE TERM;CLOSE;
ELSE
TOPSTACK:
HD(HD(FNSTACK));
CLOSE;
TL(FNSTACK)->FNSTACK;
END;

COMMENT 'THIS FUNCTION RETURNS TRUE IFF ITS OK TO EXPAND
A RECURSIVE CALL.';

FUNCTION COMPLEXITY TERM;
IF ATOM(TERM)
    THEN
    IF TERM=T THEN 66;
    ELSEIF ISNUMBER(TERM) THEN 3*TERM; ELSE 0;CLOSE;
ELSEIF HD(TERM)="QUOTE"
    THEN DATALENGTH(HD(TL(TERM)))*78+3;
    ELSE
    COMPLEXITY(HD(TERM))+COMPLEXITY(TL(TERM))+1;
    CLOSE;

```

```

END;

FUNCTION RECUROR;
VARS CMLX L HDTERM;
HD(TERM)->HDTERM;
HD(TL(HD(FNSTACK)))->CMLX;
TL(FNSTACK)->L;
LOOPIF L/=NIL
  THEN
    IF HD(HD(HD(L)))=HDTERM AND EQUAL(TERM,HD(HD(L)))
      OR CMLX>HD(TL(HD(L)))
      THEN 1->HD(TL(TL(HD(L))));0;EXIT;
    TL(L)->L;
  CLOSE;
1;
END;

```

```

FUNCTION SYMEVAL;
NIL->ALIST;
NIL->FNSTACK;
NIL->TOPLEXP;
SYMEVAL1();
END;

```

```

0->ININDUCT;

```

COMMENT THE FOLLOWING FUNCTION TRANSFORMS A SYMBOLIC LISP EXPRESSION INTO THE REAL POP-2 LIST IT REPRESENTS (IF POSSIBLE). IT IS USED ONLY FOR DEMO PURPOSES, TO MAKE EVAL LOOK LIKE ITS THE REAL THING.;

```

FUNCTION MVAL TERM;
IF ATOM(TERM)
  THEN
    IF ISVALUE(TERM) THEN TERM;1; ELSE 0;CLOSE;
  EXIT;
IF HD(TERM)="QUOTE" THEN HD(TL(TERM));1;EXIT;
IF HD(TERM)="CONS"
  THEN
    IF MVAL(HD(TL(TERM)))
      THEN
        IF MVAL(HD(TL(TL(TERM))))
          THEN
            ->TERM;
            IF ATOM(TERM)
              THEN
                IF TERM=NIL OR TERM=0 THEN NIL;
                ELSEIF ISNUMBER(TERM)
                  THEN
                    NIL->F001;
                    LOOPIF TERM THEN NIL::F001->F001;TERM-1->TERM;CLOSE;
                    F001;
                ELSE NIL::ERASE(MVAL(PRINTNAME(TERM)));CLOSE;

```

```
        ELSE TERM;CLOSE;
        CONSPAIR();1;
        EXIT;
        ERASE();
        CLOSE;
    CLOSE;
0;
END;
```

```
FUNCTION DEMOEVAL TERM;
SYMEVAL(TERM)->TERM;
IF MVAL(TERM) THEN ELSE PRSTRING('*SYMBOLIC*');NL(1);TERM;CLOSE;
END;
```