

[FOO] TRACK 63  
CREATED 16.28 1 11 1973 NEW

[ 16.28 2 NOV 1973]

```

VARS TYPE;
FUNCTION TYPE N1;SUBSRC(N1+1,
'222222222234333400403435643363430111111111111111111111111111111163434'
);
END;
FUNCTION ITR;
VARS CNT;
0->CNT;
IF C1=17 OR C1=16 OR C1=63 THEN
  1->CNT; LOOPSP: DDF1()->C2; IF C1=C2 THEN CNT+1->CNT;
  GOTO LOOPSP; CLOSE; CONSPAIR(C1,CNT); C2->C1;EXIT;

IF C1=23 THEN 1->CNT;
  APPLY(INCHARITEM(LAMBDA; IF CNT THEN 23; 0->CNT;
ELSE DDF1()->C1;C1;CLOSE;END));
EXIT;

IF C1=TERMIN THEN TERMIN;EXIT;

IF C1<10 THEN
  LOOPIF C1<10 THEN C1;DDF1()->C1;CNT+1->CNT;CLOSE;
  ISNUMBER->KIND;
  GOTO GOBBLE;
ELSEIF TYPE(C1)=1 THEN
  LOOPIF TYPE(C1)=1 OR TYPE(C1)=2 THEN C1;DDF1()->C1;CNT+1->CNT;CLOSE;
  GOTO OUTITEM;
ELSEIF TYPE(C1)=3 THEN
  LOOPIF TYPE(C1)=3 THEN C1;DDF1()->C1;CNT+1->CNT;CLOSE;
  OUTITEM:
  IF CNT<9 THEN CONSWORD(CNT);EXIT;
  ISWORD->KIND;GOTO GOBBLE;
ELSE DDF1()->C2;
  IF C2=21 THEN
    IF C1=24 OR C1= 59 THEN CONSWORD(C1,C2,2);DDF1()->C1;EXIT;
    CONSWORD(C1,1);C2->C1;RETURN;
  ELSEIF C1=21 THEN
    IF C2=25 OR C2=61 THEN CONSWORD(C1,C2,2);DDF1()->C1;EXIT;
    CONSWORD(C1,1);C2->C1;RETURN;
  ELSE CONSWORD(C1,1);C2->C1;EXIT;
CLOSE;
GOBBLE: INITC(CNT)->FOO;
LOOPIF CNT THEN ->SUBSRC(CNT,FOO);CNT-1->CNT;CLOSE;
CONSPAIR(KIND,FOO);
END;

FUNCTION INDENT DDF1 DDF2;
VARS OLD NEW;
[0] -> PDL;
```

```

DDF1() -> C1;
0 -> LINEL;
FUNCTION CUCHAROUT X;
IF X = 17 OR X = 63 THEN 0;
ELSE LINEL+1;CLOSE;
-> LINEL;
DDF2(X);
END;
CONSPAIR(16,0)->NEW;
LOOP:
NEW -> OLD;
ITR() -> NEW;
IF NEW = TERMIN THEN DDF2(TERMIN); EXIT;
IF NOT(ATOM(NEW)) AND ISNUMBER(FRONT(NEW))
  THEN
IF FRONT(NEW) = 16 THEN OLD->NEW;
ELSE PR1(NEW);CLOSE;
  GOTO LOOP;
  CLOSE;
IF LINEL = 0 THEN
  IF NEW="ELSEIF" THEN HD(PDL)-3 ;
ELSEIF NEW="COMMENT" THEN 0;
ELSE HD(PDL);CLOSE;SP();
ELSEIF SEPARATE(OLD,NEW) THEN SP(1); CLOSE;
IF NEW = "VARS" THEN VARSIND();
ELSEIF NEW = "" THEN QUOTEIND();
ELSEIF NEW = "[" THEN LISTIND();
  ELSE
  IF ISOPEN(NEW) THEN CONS(LINEL+(,),PDL) -> PDL;
  ELSEIF ISCLOSE(NEW) THEN TL(PDL) -> PDL; CLOSE;
  PR1(NEW);
  CLOSE;
GOTO LOOP;
END;

```

```

FUNCTION PR1 X;
IF ISWORD(X) THEN PR(X);
ELSEIF DATAWORD(X) = "CSTRIP" THEN PR(X);
ELSE
  IF ISNUMBER(FRONT(X))
  THEN
  IF FRONT(X) = 16 THEN SP(BACK(X));
  ELSEIF FRONT(X) = 17 THEN NL(BACK(X));
  ELSE CUCHAROUT(63); CLOSE;
  ELSE PRSTRING(BACK(X)); CLOSE;
  CLOSE;
END;

```

```

FUNCTION ISOPEN X;
VARS L;
IF ISWORD(X)
  THEN
  OPENLIST -> L;
  LOOPIF L /= NIL
  THEN IF HD(HD(L)) = X THEN HD(TL(HD(L))); 1; EXIT; TL(L)->L; CLOSE;
  0;
ELSEIF NOT(ATOM(X))
  THEN
  BACK(X) -> X;

```

```

IF DATAWORD(X) /= 9 THEN 0; EXIT;
FORALL L 1 1 9;
  IF SUBSRC(L,'OPERATION\') = SUBSRC(L,X) THEN 0; EXIT;
  CLOSE;
  0; 1;
  ELSE 0; CLOSE;
END;

```

```

FUNCTION ISCLOSE X;
  VARS L;
  CLOSELIST -> L;
  LOUPIF L /= NIL
  THEN
    IF HD(L) = X THEN 1; EXIT;
    TL(L) -> L;
  CLOSE;
  0;
END;
[[[ 1]][(% 2)][[% 2]][FUNCTION 0][IF 3][LOUPIF 3][LAMBDA 0]
 [FORALL 0][OPERATIO 0][MACRO 0]] -> OPENLIST;
[ CLOSE ) %) %] EXIT END] -> CLOSELIST;

```

```

FUNCTION VARSIND;
  PR(NEW);
  LOOP;
  NEW -> OLD;
  ITR() -> NEW;
  IF NEW = ";" THEN PR(NEW); EXIT;
  IF NOT(ATOM(NEW)) AND ISNUMBER(FRONT(NEW))
  THEN
    IF FRONT(NEW) = 17 THEN NL(1);
  ELSE OLD->NEW; CLOSE;
  GOTO LOOP;
  CLOSE;
  IF LINEL = 0 THEN SP(5);
  ELSEIF SEPARATE(OLD,NEW) THEN SP(1); CLOSE;
  PR1(NEW);
  GOTO LOOP;
END;

```

```

FUNCTION QUOTEIND;
  PR("");PR(ITR());PR("");
  ERASE(ITR());
END;

```

```

FUNCTION LISTIND;
  VARS CNT;
  PR(NEW);
  1 -> CNT;
  LOUPIF CNT
  THEN
    ITR() -> NEW;
    IF NEW = "[" THEN CNT+1->CNT;
    ELSEIF NEW = "]" THEN CNT-1->CNT; CLOSE;
    PR1(NEW);
  IF LINEL=0 THEN SP(HD(PDL));CLOSE;

```

```
CLOSE;  
END;
```

```
FUNCTION FIRSTCHAR X;  
IF ISWORD(X) THEN CHARWORD(X,1);  
ELSEIF NOT(ATOM(X)) THEN  
IF ISNUMBER(FRONT(X)) THEN FRONT(X);EXIT;  
SUBSCRC(1,BACK(X));  
ELSE 23; CLOSE;  
END;
```

```
FUNCTION SEPARATE OLD NEW;  
VARS O N;  
FIRSTCHAR(OLD) -> O;  
FIRSTCHAR(NEW) -> N;  
IF NEW=";" THEN 0;  
ELSEIF SYNTAXY(NEW) OR SYNTAXY(OLD) THEN 1;  
ELSEIF TYPE(N)=2 AND TYPE(O)=1 THEN 1;  
ELSEIF TYPE(N)=1 AND TYPE(O)=2 THEN 1;  
ELSEIF TYPE(N)>3 OR TYPE(O)>3 THEN 0;  
ELSE TYPE(N)=TYPE(O);CLOSE;  
  
END;
```

```
FUNCTION SYNTAXY X;  
IF X="LOOPIF" OR X="THEN" OR X="OR" OR X="AND" OR X="ELSE" OR  
X="IF" OR X="ELSEIF" THEN 1;  
ELSE 0;CLOSE;  
  
END;
```

```
FUNCTION PPRLP80 FN;  
VARS DDF1 DDF2;  
DIN(FN)->DDF1;  
POPMESS("LP80":FN)->DDF2;  
INDENT(DDF1,DDF2);  
END;
```

```
FUNCTION PPRCOPY FN1 FN2;  
INDENT(DIN(FN1),DOUT(FN2));  
END;
```

```
FUNCTION PPRFILE FN1;  
PPRCOPY(FN1,FN1);  
END;
```

```
LAMBDA;  
IF CUCHAROUT=CHAROUT THEN  
'PPRLP80(FN)=>();  
PPRCOPY(FN1,FN2)=>();  
PPRFILE(FN1)=>();  
' .PRSTRING;  
CLOSE;  
END;  
.APPLY;
```

[POPTRANS] TRACK 63  
CREATED 12.51 30 10 1973

[ 16.28 2 NOV 1973]

```
FUNCTION CAR X;  
IF ATOM(X) THEN NIL; ELSE HD(X); CLOSE;  
END
```

```
FUNCTION CDR X;  
IF ATOM(X)  
  THEN  
    IF X = NIL OR NOT(X) THEN NIL;  
    ELSEIF ISNUMBER(X) THEN X-1;  
    ELSE [% ERASE(DESTWORD(X)) %]; CLOSE;  
    ELSE TL(X); CLOSE;  
END;
```

```
FUNCTION CONS X Y;  
IF X /= NIL AND X  
  THEN CONSPAIR(X,Y);  
ELSEIF ISNUMBER(Y)  
  THEN Y+1;  
  ELSE  
    0 -> X;  
    Y -> F001;  
    LOOPIF Y /= NIL AND ISNUMBER(HD(Y)) AND X < 8  
      THEN HD(Y);X+1->X;TL(Y)->Y;CLOSE;  
    IF Y = NIL THEN CONSWORD(X);  
    ELSE LOOPIF X THEN ERASE();X-1->X;CLOSE;CONSPAIR(NIL,F001);CLOSE;  
    CLOSE;  
END;
```

```
FUNCTION EQUAL X Y;  
TOP:  
IF ATOM(X) OR ATOM(Y)  
  THEN  
    IF X = Y THEN T;  
    ELSEIF X=NIL AND NOT(Y) THEN T;  
    ELSEIF NOT(X) AND Y=NIL THEN T;  
    ELSE NIL; CLOSE;  
ELSEIF EQUAL(HD(X),HD(Y))  
  THEN HD(X)->X;HD(Y)->Y;GOTO TOP;  
  ELSE NIL; CLOSE;  
END;
```

```
FUNCTION POPTRANS TERM JUMPOK;  
VARS JUMP ARGS CHNGDARGS ELSEIFFLAG;  
IF ATOM(TERM) THEN TERM; EXIT;  
LISPHASH(HD(TERM)) SWITCH FNAPP FNAPP QUOTE LISPCONS  
  LISPEQUAL LISPIF FNAPP;  
QUOTE: """; HD(TL(TERM)); """; RETURN;
```

```
LISPCONS: "CONS" :: TL(TERM) -> TERM; GOTO FNAPP;
```

```
LISPEQUAL: "EQUAL" :: TL(TERM) -> TERM; GOTO FNAPP;
```

```

LISPIF:
IF STACKLEN() AND (->FOO1; FOO1 = "ELSE")
  THEN "ELSEIF"; 1;
  ELSE FOO1; "IF"; 0; CLOSE;
-> ELSEIFFLAG;
IF BOOLEAN(HD(TL(TERM)))
  THEN
  POPTRANS(HD(TL(TERM)),0); "="; T;
  ELSE
  IF ATOM(HD(TL(TERM)))
    THEN HD(TL(TERM));
    ELSE
    ERASE();
    POPTRANS(HD(TL(TERM)),0); "->"; "TESTTEMP"; ";";
    1 -> NEEDTEMP;
    IF ELSEIFFLAG THEN "ELSEIF"; ELSE "IF"; CLOSE;
    "TESTTEMP";
    CLOSE;
  -> FOO1;
  FOO1; "/="; NIL; "AND"; FOO1;
  CLOSE;
"THEN"; POPTRANS(HD(TL(TL(TERM))),JUMPOK); ";";
"ELSE"; POPTRANS(HD(TL(TL(TL(TERM)))),JUMPOK);
IF ELSEIFFLAG THEN ELSE ";"; "CLOSE"; CLOSE;
RETURN;

```

```

FNAPP:
IF HD(TERM) = FNSYMB AND JUMPOK
  THEN 1;
  ELSE HD(TERM); "("; 0; CLOSE;
-> JUMP;
ARGLIST -> ARGS;
NIL -> CHNGDARGS;
LOOPIF (TL(TERM)->TERM; TERM /= NIL)
  THEN
  IF JUMP
    THEN
    HD(ARGS) -> FOO1;
    TL(ARGS) -> ARGS;
    IF FOO1 = HD(TERM) THEN GOTO SKIP; CLOSE;
    FOO1 :: CHNGDARGS -> CHNGDARGS;
    CLOSE;
  POPTRANS(HD(TERM),0); ", ";
SKIP:
CLOSE;
ERASE();
IF JUMP
  THEN
  1 -> PUTTABLE;
  IF TL(CHNGDARGS) /= NIL THEN ";"; CLOSE;
  LOOPIF CHNGDARGS /= NIL THEN "->"; HD(CHNGDARGS);
  TL(CHNGDARGS) -> CHNGDARGS; CLOSE; ";";
  "GOTO"; "TOP";
  ELSE ")"; CLOSE;

```

END;

```

FUNCTION POP2OF X;
VARS FNSYMB ARGLIST PUTTABLE NEEDTEMP;

```

```
HD(X) -> FNSYMB;
HD(TL(HD(TL(X)))) -> ARGLIST;
SIMPLIFY(HD(TL(TL(HD(TL(X)))))) -> X;
0 -> PUTTABLE;
0 -> NEEDTEMP;
[% POPTRANS(X,1), ";", "END", ";" %] -> X;
[% "FUNCTION", FNSYMB, APPLIST(ARGLIST,IDENTFN), ";",
  (IF NEEDTEMP THEN "VARS"; "TESTTEMP"; ";;"; CLOSE),
  (IF PUTTABLE THEN "TOP"; ";;"; CLOSE) %] <> X;
END;
```

```
FUNCTION DO L;
VARS X;
LOOPIF L /= NIL
  THEN
  POP2OF([% HD(L), PROP("DEFN",HD(L)) %]) -> X;
  PPRPOPLISP(X);
  POPVAL(X<>[GOON]);
  NL(2);
  TL(L)->L;
  CLOSE;
END;
```

[PPRPOPLI] TRACK 63  
CREATED 11.39 30 10 1973

[ 16.29 2 NOV 1973]

```
FUNCTION PRRPOPLISP POPLISP;
VARS X OLDX INDENT;
0 -> INDENT;
"(" -> X;
LOOPIF POPLISP /= NIL
THEN
X -> OLDX;
HD(POPLISP) -> X;
TL(POPLISP) -> POPLISP;
INDENT + INDENTINCR(X) -> INDENT;
IF NEWLINEA(OLDX) OR NEWLINEB(X) THEN NL(1); SP(INDENT);
ELSEIF SEPARATE(OLDX,X) THEN SP(1); CLOSE;
PR(X);
IF X = ""
THEN PR(HD(POPLISP));PR(X);TL(TL(POPLISP)) -> POPLISP; CLOSE;
CLOSE;
END;

MEMBER(% [; :] %) -> NEWLINEA;
MEMBER(% [THEN ELSE CLOSE END FUNCTION VARS IF ELSEIF] %)
-> NEWLINEB;

FUNCTION INDENTINCR X;
IF MEMBER(X,OPENLIST) THEN +3;
ELSEIF MEMBER(X,CLOSELIST) THEN -3;
ELSE 0; CLOSE;
END;

[THEN] -> OPENLIST;
[ELSEIF CLOSE] -> CLOSELIST;

FUNCTION SEPARATE X Y;
IF ASSOC(X,XSEPALIST)
THEN HD(BACK());
IF ASSOC(Y,YSEPALIST) THEN LOGAND(HD(BACK())); CLOSE;
ELSEIF ASSOC(Y,YSEPALIST) THEN HD(BACK()); ELSE 1; CLOSE;
END;

[[ ( 0] [) 1] [, 0] [-> 1] [/= 1] [= 1] [" 1]] -> XSEPALIST;
[[ : 0] [; 0] [( 0] [) 0] [, 0] [-> 1] [/= 1] [= 1] [" 1]] -> YSEPALIST;
```



[ / USERINDU] TRACK 63  
CREATED 12.45 25 10 1973

[ 16.29 2 NOV 1973]

PICKINDCONST -> SYSINDUCT;

```
FUNCTION USERINDUCT INDTERM;
  VARS ARGLIST RECPOCKETS OTHERFAILS;
  POPTTON();
  NL(4);
  IF SYSINDUCT(INDTERM)
  THEN
    -> ARGLIST -> RECPOCKETS -> OTHERFAILS;
    PRSTRING('I WOULD INDUCT ON THE FOLLOWING: ');PR(ARGLIST);NL(2);
    PRSTRING('HYPOTHESES WOULD BE SUPPLIED FOR THE FOLLOWING COMBINATIONS:
  ');PPR(RECPOCKETS);NL(2);
    PRSTRING('IN ADDITION, THE CONCLUSION WOULD EXPLICITLY CONTAIN THE
  FOLLOWING
  SUBSTRUCTURES: ');PR(OTHERFAILS);NL(2);
    PRSTRING('IS THAT OKAY (Y/N)');
    IF CHARIN() = 57 THEN OTHERFAILS;RECPOCKETS;ARGLIST;1;EXIT;
    ELSE
    PRSTRING('I CANNOT FIND ANYTHING TO INDUCT ON. ');NL(2);
    CLOSE;
    PRSTRING('TYPE THE LIST OF SKOLEM CONSTANTS TO BE INDUCTED UPON:
  ');LISTREAD()->ARGLIST;
    PRSTRING('TYPE THE LIST OF COMBINATIONS OF SUBSTRUCTURES FOR WHICH
  HYPOTHESES WILL BE SUPPLIED: ');LISTREAD()->RECPOCKETS;
    PRSTRING('TYPE THE LIST OF ADDITIONAL SUBSTRUCTURES TO BE EXPLICITLY
  MENTIONED IN THE CONCLUSION: ');LISTREAD()->OTHERFAILS;
    NL(2);
    OTHERFAILS;RECPOCKETS;ARGLIST;1;
  END;

PRSTRING('TO MANUALLY SPECIFY WHAT TO INDUCT UPON, EXECUTE:
  USERINDUCT -> PICKINDCONSTS;
  BEFORE STARTING THE PROOF. TO LET THE SYSTEM DO IT, USE:
  SYSINDUCT -> PICKINDCONSTS;
  THE DEFAULT IS SYSINDUCT.
  ');
```

CREATED 12.55 24 10 1973

COMMENT 'THIS IS THE NORMALIZE FUNCTION. IN-LINE COMMENTS EXPLAIN THE REWRITE RULES APPLIED.';

VARS REWRITEFN;

IDENTFN -> REWRITEFN;

FUNCTION REWRITE TERM;  
VARS TERM1 TERM2 TERM3;

IF HD(TERM) = "USED"  
THEN  
IF SHD(HD(TL(TERM))) = "IF" OR ISSPEC(HD(TL(TERM)))  
THEN HD(TL(TERM));ELSE TERM;CLOSE;  
EXIT;

COMMENT 'IF TERM IS AN EQUALITY';

IF HD(TERM)="EQUAL" THEN  
HD(TL(TERM))->TERM1;  
HD(TL(TL(TERM)))->TERM2;

COMMENT '(EQUAL KNOWN1 KNOWN2) => T OR NIL';  
IF TERM1 == TERM2 THEN T; EXIT;  
IF NOTIDENT THEN NIL; EXIT;

COMMENT '(EQUAL BOOL T) => BOOL';  
IF TERM1=T AND BOOLEAN(TERM2)THEN TERM2 EXIT;  
IF TERM2=T AND BOOLEAN(TERM1) THEN TERM1 EXIT;

COMMENT '(EQUAL (EQUAL A B) C) =>  
(IF (EQUAL A B) (EQUAL C T) (IF C NIL T))';  
IF SHD(TERM1) = "EQUAL" OR SHD(TERM2) = "EQUAL" AND (SWAP;1)  
THEN  
[% "IF", TERM1,  
REWRITE([% "EQUAL", TERM2, T %]),  
REWRITE([% "IF", TERM2, NIL, T %]) %] -> TERM;  
GOTO CONDL;  
CLOSE;

COMMENT '(EQUAL X NIL) => (IF X NIL T)';  
IF TERM1 == NIL OR TERM2 == NIL AND (SWAP;1)  
THEN  
[% "IF", TERM2, NIL, T %] -> TERM;  
GOTO CONDL;  
CLOSE;

COMMENT 'GO SEE IF ONE ARG IS A IF';  
GOTO CONDARG;

```
COMMENT 'TERM IS A IF';
```

```
ELSEIF HD(TERM)="IF" THEN
```

```
CONDL:
```

```
TL(TERM)->TERM3;
```

```
HD(TERM3)->TERM1;
```

```
TL(TERM3)->TERM3;
```

```
HD(TERM3)->TERM2;
```

```
HD(TL(TERM3))->TERM3;
```

```
COMMENT '(IF KNOWN X Y) => X OR Y';
```

```
IF TERM1 == NIL THEN TERM3; EXIT;
```

```
IF NOTIDENT THEN TERM2; EXIT;
```

```
COMMENT '(IF X Y Y) => Y';
```

```
IF TERM2 == TERM3 THEN TERM2; EXIT;
```

```
COMMENT '(IF X X NIL) => X';
```

```
IF TERM1 == TERM2 AND TERM3 == NIL THEN TERM1; EXIT;
```

```
COMMENT '(IF BOOL T NIL) => BOOL';
```

```
IF BOOLEAN(TERM1) AND TERM2 = T AND TERM3 == NIL
```

```
THEN TERM1; EXIT;
```

```
COMMENT '(IF X T (IF Y NIL T)) => (IF Y (IF X T NIL) T)';
```

```
IF TERM2=T AND SHD(TERM3)="IF" AND
```

```
HD(TL(TL(TERM3))) == NIL AND HD(TL(TL(TL(TERM3)))) = T
```

```
THEN
```

```
IF BOOLEAN(TERM1)
```

```
THEN TERM1;
```

```
ELSE [% "IF", TERM1, T, NIL %] CLOSE;
```

```
-> TERM2;
```

```
HD(TL(TERM3)) -> TERM1;
```

```
T -> TERM3;
```

```
[% "IF", TERM1, TERM2, TERM3 %] -> TERM;
```

```
CLOSE;
```

```
COMMENT 'IF TERM1 IS AN IF, DECIDE IF IT SHOULD BE  
DISTRIBUTED.';
```

```
IF SHD(TERM1) = "IF" THEN
```

```
COMMENT '(IF (IF A T2 T3) B C) => (IF A (IF T2 B C)
```

```
(IF T3 B C)) WHERE T2 OR T3 ISNIL';
```

```
IF HD(TL(TL(TERM1))) == NIL OR HD(TL(TL(TL(TERM1)))) == NIL
```

```
THEN
```

```
GOTO CONDCOND;
```

```
CLOSE;
```

```
COMMENT '(IF (IF A T (* N)) T (* M)) => (IF A T (* N M))';
```

```
IF TERM2 = T AND SHD(TERM3) = "*" AND HD(TL(TL(TERM1))) = T
```

```
AND SHD(HD(TL(TL(TL(TERM1)))))) = "*" 
```

```
THEN
```

```
[% "IF", HD(TL(TERM1)), T, "*" :: (TL(HD(TL(TL(TL(TERM1))))))
```

```

        <> TL(TERM3)) %J];
EXIT;

COMMENT '(IF (IF A B C) D E)=> (IF A (IF B C E) (IF C D E))
        WHERE D AND E ARE NOT NIL OR D AND E ARE T AND NIL';

IF TERM2 == NIL AND TERM3 /= T THEN GOTO SKIP;
ELSEIF TERM3 == NIL AND TERM2 /= T THEN GOTO SKIP; CLOSE;
CONDCOND:
    IF SHD(TERM2) = "*" OR SHD(TERM3) = "*" THEN GOTO SKIP; CLOSE;
    REWRITE([%"IF", HD(TL(TL(TERM1))), TERM2,TERM3 %J]);
    REWRITE([%"IF", HD(TL(TL(TL(TERM1)))),TERM2,TERM3%J]);
    ->TERM3->TERM2;
    [%"IF",HD(TL(TERM1)),TERM2,TERM3%J]->TERM;
    GOTO CONDL;
    SKIP:
CLOSE;

COMMENT 'TERM IS A NON-IF, NON-EQ FUNCTION CALL';
ELSE

COMMENT '(FOO X (IF A B C) Y) =>
        (IF A (FOO X B Y) (FOO X C Y))';

CONDARG:
TL(TERM) -> TERM1;
LOOPIF TERM1 /= NIL AND SHD(HD(TERM1)) /= "IF"
    THEN
    TL(TERM1) -> TERM1;
    CLOSE;
    IF TERM1 /= NIL
    THEN
    HD(TERM1) -> TERM1;
    [%"IF", HD(TL(TERM1)), REWRITE(SUBST(HD(TL(TL(TERM1))),TERM1,
    TERM)), REWRITE(SUBST(HD(TL(TL(TL(TERM1)))),TERM1,TERM)) %]
    -> TERM;
    GOTO CONDL;
    CLOSE;
CLOSE;
REWRITEFN();
TERM
END

FUNCTION NORMALIZE TERM;
IF ATOM(TERM)THEN TERM EXIT;
REWRITE(HD(TERM)::MAPLIST(TL(TERM),NORMALIZE));
END

```

[ / USE ] TRACK 63  
CREATED 12.49 24 10 1973

[ 16.29 2 NOV 1973 ]

DCOMP([ / USE FERTILIZE ] );  
DCOMP([ / USE REWRITE ] );  
DCOMP([ / USE REDUCE ] );  
DCOMP([ / USE PROVE ] );

[ / UNDO USE] TRACK 63  
CREATED 20.07 22 10 1973

[ 16.29 2 NOV 1973]

DTRACK(22);  
APPLIST([[ /REWRITE][ /REDUCE][ /FERTILIZE]],DCOMP);  
DTRACK(36);  
DCOMP([ /PROVE]);

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 REMOVES USED TERMS AND  
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;  
0 -> CONJFLAG;  
IF FERTILIZE(THM)  
THEN 1; EXIT;  
REMUSED(THM) -> THM;  
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;      (REPORT("/", IDENTFN, "PROVE1"));
  LOOP:
  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;
        POPTTON(); NL(10); PR(POPDATE()); NL(3);
        APPLIST(GETTHM(THM),
            LAMBDA THM;
            IF AVOIDSTAR AND MEMBER("*", HD(THM)) THEN EXIT;
            PROVE1(THM); TOTTIME+THMTIME->TOTTIME; END);
        POPTTON();
        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 -> CCHAROUT;
        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;

```

```

DEMOEVAL -> RLOOPFN;

```

```
FUNCTION REDUCE1 TERM CONSALIST;
VARS TERM1 TERM2 TERM3 USEDTERM1;
RECURSE:
COMMENT 'IF TERM IS ATOM OR NON-IF, QUIT';
IF ATOM(TERM) OR HD(TERM) /= "IF"
  THEN
    TERM;
  EXIT;

COMMENT 'GET COMPONENTS OF THE IF';
HD(TL(TERM)) -> TERM1;
HD(TL(TL(TERM))) -> TERM2;
HD(TL(TL(TL(TERM)))) -> TERM3;

COMMENT 'SKIP OVER USED MARKER BUT REMEMBER IT';
IF SHD(TERM1) = "USED"
  THEN HD(TL(TERM1)) -> TERM1; 1;
  ELSE 0; CLOSE;
-> USEDTERM1;

COMMENT 'IF TERM1 IS NIL OR CONS, EVAL IT';
IF TERM1 == NIL
  THEN
    TERM3 -> TERM;
    GOTO RECURSE;
  ELSEIF EXPLCONS(TERM1)
    THEN
      TERM2 -> TERM;
      GOTO RECURSE;
  ELSEIF ASSOCID(TERM1,CONSALIST)
    THEN
      -> F001;
      1 -> BACK(F001);
      TERM2 -> TERM;
      GOTO RECURSE;
    CLOSE;

COMMENT '(IF ATOM A B) => (IF ATOM R(A(ATOM/CONS)) R(B(ATOM/NIL)))';
IF ATOM(TERM1)
  THEN
    GOTO SUBSTCONS;
  CLOSE;

COMMENT '(IF (EQUAL A SPECLIST) B C) => (IF (EQUAL A SPECLIST)
  R(B(A/SPECLIST))
  R(C((EQUAL A SPECLIST)/NIL)))';
IF HD(TERM1) = "EQUAL"
  THEN
    IF ISSPEC(HD(TL(TERM1)))
      THEN SUBST(HD(TL(TERM1)),HD(TL(TL(TERM1))),TERM2);
    ELSEIF ISSPEC(HD(TL(TL(TERM1))))
      THEN SUBST(HD(TL(TL(TERM1))),HD(TL(TERM1)),TERM2);
```

```

ELSE GOTO SUBSTRUE; CLOSE;

GOTO TESTSUBST;
CLOSE;

COMMENT '(IF (IF ...) A B) => (IF R(IF) R(A) R(B))';
IF HD(TERM1) = "IF"
  THEN
    REDUCE1(TERM1,CONSALIST) -> TERM1;
    REDUCE1(TERM2,CONSALIST) -> TERM2;
    REDUCE1(TERM3,CONSALIST) -> TERM3;
    IF TERM3 == NIL THEN GOTO CONTINUE; CLOSE;
    [% "IF", TERM1, TERM2, TERM3 %];
    EXIT;

CONTINUE:

COMMENT '(IF BOOL A B) => (IF BOOL R(A(BOOL/T)) R(B(BOOL/NIL)))';
IF BOOLEAN(TERM1)
  THEN
    SUBSTRUE:
      SUBST(T,TERM1,TERM2);
    COMMENT 'NOW DECIDE IF SUBST INTO TERM2 HAD EFFECT. IF
    SO, MARK TERM1 AS "USED" WHEN THE ANSWER IS CONSED UP.';
    TESTSUBST:
      -> F001;
      IF USEDTERM1 THEN ELSE (F001 /= TERM2) -> USEDTERM1;CLOSE;
      REDUCE1(F001,CONSALIST) -> TERM2;
      REDUCE1(SUBST(NIL,TERM1,TERM3),CONSALIST) -> TERM3;
      IF USEDTERM1
        THEN
          [% "USED", TERM1 %] -> TERM1;
          CLOSE;
          [% "IF", TERM1, TERM2, TERM3 %];
          EXIT;

COMMENT '(IF RANDOM A B) => (IF RANDOM R(A(RANDOM/CONS))
                                R(B(RANDOM/NIL)))';
SUBSTCONS:
CONSPAIR(TERM1,0) -> TERM1;
REDUCE1(TERM2,TERM1::CONSALIST) -> TERM2;
REDUCE1(SUBST(NIL,FRONT(TERM1),TERM3),CONSALIST) -> TERM3;
IF USEDTERM1 OR BACK(TERM1)
  THEN
    [% "USED", FRONT(TERM1) %];
    ELSE FRONT(TERM1); CLOSE;
-> TERM1;
[% "IF", TERM1, TERM2, TERM3 %];

END;

1 -> PROP("BOOLEAN","USED");

REDUCE1(% NIL %) -> REDUCE;

COMMENT 'THIS FUNCTION REMOVES TERMS MARKED "USED". IT

```

HIDES THOSE IN HYP POSITIONS AND SIMPLY UN-MARKS ALL OTHERS.';

```
FUNCTION REMUSED1 TERM;
VARS TERM1 TERM2 TERM3;
IF ATOM(TERM) THEN TERM; RETURN;
ELSEIF HD(TERM) = "USED" THEN HD(TL(TERM)); RETURN;
ELSEIF HD(TERM) = "IF"
  THEN
  HD(TL(TERM)) -> TERM1;
  HD(TL(TL(TERM))) -> TERM2;
  HD(TL(TL(TL(TERM)))) -> TERM3;
  IF SHD(TERM1) = "USED"
    THEN
    IF TERM3 /= NIL AND TERM3 AND SHD(TERM3) /= "*"
      AND BOOLEAN(TERM2) AND BOOLEAN(TERM3)
      THEN
      [% "IF", REMUSED1(TERM2),
        T,
        GENSTAR(["IF",HD(TL(TERM1)),NIL,TERM3%])%]
      -> TERM2;
    REMUSED1(TERM3) -> TERM3;
    IF TERM3 = T
      THEN TERM2;
    ELSE [% "IF", TERM2,
          [% "IF", TERM3, T, HD(TL(TERM1)) %],
          NIL %]; CLOSE;
    EXIT;
  CLOSE;
  CLOSE;
  [% HD(TERM), APPLIST(TL(TERM),REMUSED1) %];
END;
```

```
FUNCTION REMUSED THM;
VARS OLDCNT;
STARCOUNT -> OLDCNT;
REMUSED1(THM) -> THM;
IF STARCOUNT /= OLDCNT
  THEN
  NORMALIZE(THM) -> THM;
  REPORT(["R",STARCOUNT-OLDCNT %], 'REMOVING USED TERMS PRODUCES:',
    "REMUSED");
  CLOSE;
THM;
END;
```

CREATED NIL

```

FUNCTION FERTREPORT;
IF VERBOSE
  THEN
    POPTTON();
    NL(4);PRSTRING('FERTILIZE WITH ');ppRIND(TERM1,15,1);
    PRSTRING('.');NL(2);
    CLOSE;
END;

```

```

FUNCTION FERTILIZE TERM;
VARS TERM1 TERM2 TERM3 LHS1 RHS1;
IF SHD(TERM) /= "IF" THEN 0; EXIT;
HD(TL(TERM)) -> TERM1;
HD(TL(TL(TERM))) -> TERM2;
HD(TL(TL(TL(TERM)))) -> TERM3;
IF SHD(TERM1) = "USED" THEN HD(TL(TERM1))->TERM1;CLOSE;

```

```

COMMENT 'LOOK FOR TERMS OF THE FORM (IF (EQUAL LHS RHS) BOOL1 BOOL2)
WHERE BOOL2 IS NOT NIL. IF FOUND, FERTILIZE
LHS=RHS INTO BOOL1 AND HIDE IT.';

```

```

IF SHD(TERM1) = "EQUAL" AND TERM3 /= NIL
  AND TERM3 AND BOOLEAN(TERM2) AND BOOLEAN(TERM3)
  THEN
    HD(TL(TERM1)) -> LHS1;
    HD(TL(TL(TERM1))) -> RHS1;
    IF ISSPEC(LHS1) OR ISSPEC(RHS1)
      THEN GOTO NOFERT; CLOSE;
    IF FERTIL1(TERM2)
      THEN
        -> TERM2;
        IF FERTILIZE(TERM2) THEN -> TERM2; CLOSE;
        [% "IF", TERM2, T, GENSTAR(["IF",TERM1,NIL,TERM3%])%]
        -> TERM2;
        REPORT(["F",STARCOUNT%],FERTREPORT,"FERTILIZE");
    IF FERTILIZE(TERM3) THEN ->TERM3; CLOSE;
    IF TERM3 = T
      THEN TERM2;
      ELSE
        [% "IF", TERM2,
          [% "IF", TERM3,
            T,
            TERM1 %],
          NIL %];
        CLOSE;
    1;
    EXIT;
    CLOSE;

NOFERT:
0 -> TERM1;

```

```

[% "IF", APPLIST(TL(TERM),
  LAMBDA TERM;
  IF FERTILIZE(TERM) THEN 1->TERM1; ELSE TERM; CLOSE;
  END),
  (IF TERM1 THEN ELSE
    ERASE(ERASE(),ERASE(),ERASE(),ERASE()); 0;EXIT) %];
1;
END;

```

```

FUNCTION FERTIL1 TERM;
  VARS LHS2 RHS2;
  IF ATOM(TERM) THEN 0; EXIT;
  IF HD(TERM) = "EQUAL"
  THEN
    HD(TL(TERM)) -> LHS2;
    HD(TL(TL(TERM))) -> RHS2;
  COMMENT 'NOW LOOK FOR (THE BEST) CROSS FERTILIZATION';
  IF OCCUR(RHS1,RHS2)
  THEN
    IF OCCUR(LHS1,LHS2)
    THEN
      IF CONSCNT(RHS1) < CONSCNT(LHS1)
      THEN SUBST(RHS1,LHS1,LHS2) -> LHS2;
      ELSE SUBST(LHS1,RHS1,RHS2) -> RHS2; CLOSE;
      ELSE SUBST(LHS1,RHS1,RHS2) -> RHS2; CLOSE;
    ELSE
      IF OCCUR(LHS1,LHS2)
      THEN SUBST(RHS1,LHS1,LHS2) -> LHS2;
      ELSE GOTO MASSSUBST; EXIT;
    CLOSE;
  [% "EQUAL", LHS2, RHS2 %];
  1;
  EXIT;
  COMMENT 'IF TERM IS AN IF, LOOK FOR ITS "CORE" AND FERTILIZE IT';
  IF HD(TERM) = "IF"
  THEN
    COMMENT '(IF X CORE T) => (IF X FERT(CORE) T), PROVIDED
  X DOES NOT CONTAIN LHS1 OR RHS1.';
    IF HD(TL(TL(TL(TERM)))) = T
    THEN
      IF OCCUR(LHS1,HD(TL(TERM))) THEN GOTO CHKRHSOCC;
      ELSEIF OCCUR(RHS1,HD(TL(TERM))) THEN GOTO SUBSTLR;
      ELSEIF FERTIL1(HD(TL(TL(TERM))))
      THEN
        -> F001;
        [% "IF", HD(TL(TERM)), F001, T %];
        1;
        EXIT;
    COMMENT '(IF CORE T (*N)) => (IF FERT(CORE) T (*N))';
    ELSEIF HD(TL(TL(TERM))) = T AND SHD(HD(TL(TL(TL(TERM))))="*"
    THEN
      IF FERTIL1(HD(TL(TERM)))
      THEN
        -> F001;
        [% "IF", F001, T, HD(TL(TL(TL(TERM)))) %];
        1;
        EXIT;
      CLOSE;
    COMMENT 'IF NOT OF EITHER OF THE ABOVE FORMS, FALL THROUGH
  TO MASSIVE SUBSTITUTION.';

```

```
CLOSE;

COMMENT 'IF CROSS FERTILIZATION NOT POSSIBLE, TRY MASSIVE
SUBSTITUTION';

MASSSUBST:
IF OCCUR(LHS1,TERM)
  THEN
CHKRHSOCC:
  IF OCCUR(RHS1,TERM)
    THEN
      IF CONSCNT(RHS1) < CONSCNT(LHS1)
        THEN SUBST(RHS1,LHS1,TERM);
      ELSE SUBST(LHS1,RHS1,TERM); CLOSE;
      ELSE SUBST(RHS1,LHS1,TERM); CLOSE;
    ELSEIF OCCUR(RHS1,TERM)
      THEN
SUBSTLH: SUBST(LHS1,RHS1,TERM);
      ELSE 0; EXIT;
1;

END;
```