

21.19HRS. 14 SEPT 1973.

** RSB **

[FILE DUMP]

[/]

[/ REWRITE]

[/ PROVE]

[/ IND1]

[/ IND2]

[/ VERB]

[/ FERTILIZ]

[/ REDUCE]

[/ GENRLIZE]

22

[/] TRACK 22
CREATED 20.05 14 9 1973

[21.19 14 SEPT 1973]

[9 22 36]->DTRS;

VARS SLASH9 SLASH22;

[[/PROPS][PPR][GEN][GENSYM][INPUT][TYPE][IDENT][EVAL]] -> SLASH9;

[[/REWRITE][REDUCE][FERTILIZE][GENRLIZE][IND1][IND2][PROVE]]
-> SLASH22;

DTRACK(9);
APPLIST(SLASH9,DCOMP);

DTRACK(22);
APPLIST(SLASH22,DCOMP);

DTRACK(36);
DCOMP([/DEFS]);
APPLIST(ALLFNS,NORMDEF);
DTRACK(22);

[/ REWRITE] TRACK 22
CREATED 10.19 1 8 1973

[21.19 14 SEPT 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;

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';
IDENT(TERM1,TERM2) -> TERM3;
IF TERM3 = NIL THEN NIL; EXIT;
IF TERM3 THEN "T";EXIT;

COMMENT '(EQUAL BOOL T) => BOOL';
IF TERM1==1 AND BOOLEAN(TERM2)THEN TERM2 EXIT;
IF TERM2==1 AND BOOLEAN(TERM1) THEN TERM1 EXIT;

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

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

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

COMMENT 'TERM IS A COND';

ELSEIF HD(TERM)="COND" THEN

COND:
TL(TERM)->TERM3;
HD(TERM3)->TERM1;

```

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

COMMENT '(COND KNOWN X Y) => X OR Y';
IF TERM1 == NIL THEN TERM3; EXIT;
IF ISCONS(TERM1) THEN TERM2; EXIT;

COMMENT '(COND X Y Y) => Y';
IF TERM2 == TERM3 THEN TERM2; EXIT;

COMMENT '(COND X X NIL) => X';
IF TERM1 == TERM2 AND TERM3 == NIL THEN TERM1; EXIT;

COMMENT '(COND BOOL T NIL) => BOOL';
IF BOOLEAN(TERM1) AND TERM2 == "T" AND TERM3 == NIL
THEN TERM1; EXIT;

COMMENT '(COND X T (COND Y NIL T)) => (COND Y (COND X T NIL) T)';
IF TERM2=="T" AND SHD(TERM3)="COND" AND
HD(TL(TL(TERM3))) == NIL AND HD(TL(TL(TL(TERM3)))) == "T"
THEN
IF BOOLEAN(TERM1)
THEN TERM1;
ELSE [% "COND", TERM1, "T", NIL %] CLOSE;
-> TERM2;
HD(TL(TERM3)) -> TERM1;
"T" -> TERM3;
[% "COND", TERM1, TERM2, TERM3 %] -> TERM;
CLOSE;

COMMENT '(COND (COND A T2 T3) B C) => (COND A (COND T2 B C)
(COND T3 B C)) WHERE T2 OR T3 ISNIL';

IF SHD(TERM1) = "COND" AND
HD(TL(TL(TERM1))) == NIL OR HD(TL(TL(TL(TERM1)))) == NIL
THEN
GOTO CONDCOND;
CLOSE;

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

IF SHD(TERM1) = "COND"
THEN
IF TERM2 == NIL AND NOT(TERM3 == "T") THEN GOTO SKIP;
ELSEIF TERM3 == NIL AND NOT(TERM2 == "T") THEN GOTO SKIP; CLOSE;
CONDCOND:
IF ISSTAR(SHD(TERM2)) OR ISSTAR(SHD(TERM3)) THEN GOTO SKIP; CLOSE;
REWRITE([%"COND", HD(TL(TL(TERM1))), TERM2, TERM3 %]);
REWRITE([%"COND", HD(TL(TL(TL(TERM1)))) , TERM2, TERM3 %]);
->TERM3->TERM2;
[% "COND", HD(TL(TERM1)), TERM2, TERM3 %]->TERM;
GOTO COND;
SKIP:
CLOSE;

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

```

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

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

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

[/ PROVE] TRACK 22
CREATED 10.53 27 7 1973

[21.2 14 SEPT 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, 0)); END);

END;

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 SUBSRCR(DATALLENGTH(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;
      POPTON(); 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);->CUCHAROUT;
      CLOSE;
    ELSE
      CHAROUT->CUCHAROUT;POPTON(); NL(1); PRSTRING('PROFILES DIVERGE');NL(1);SETPOP
      ();
      CLOSE;
      CLOSE;
      END;

```

```

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

```

```

FUNCTION SETUP;
NIL -> PROFILE;
NIL -> FERTLIST;
NIL -> GENRLLIST;
NIL -> BOMBLIST;
NIL -> PROVEFNS;
NIL -> GENALIST;
NIL -> INDVARLIST;
UNDEF -> ENDTHM;
POPTIME -> THMTIME;
IF NOT(ATOM(HD(THM)))
THEN
  NL(2);
  PR(HD(THM));
  TL(THM) -> THM;
  CLOSE;
THM -> LASTREPTHM;
THM -> LASTPPRTHM;
IF VERBOSE
THEN SP(5); PR(POPDATE()); CLOSE;
NL(1);
END;

```

```

FUNCTION FINREPORT;
POPTON();

```

```

IF VERBOSE = 1
THEN
IF PROVEFNS /= NIL
THEN
NL(5);PRSTRING('FUNCTION DEFINITIONS:');NL(2);
APPLIST(REV(PROVEFNS),
LAMBDA FN;
IF ISSTAR(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 FERTLIST /= NIL
THEN
NL(3);PRSTRING('FERTILIZERS:');NL(2);
APPLIST(REV(FERTLIST),
LAMBDA X;
PR(X);PRSTRING(' = ');PPRIND(PROP("AUXDEFN",X),5,0);NL(2);
END);
CLOSE;
IF GENRLLIST /= NIL
THEN
NL(3); PRSTRING('GENERALIZATIONS:');NL(2);
APPLIST(REV(GENRLLIST),
LAMBDA X;
PR(BACK(X)); PRSTRING(' = '); PR(FRONT(X));NL(2);
END);
CLOSE;
CLOSE;
IF VERBOSE
THEN
NL(3);
PRSTRING('PROFILE: ');PR(REV(PROFILE));NL(2);
CLOSE;
END;

```

```

FUNCTION WRAPUP;
IHM -> ENDTHM;
(POPTIME - THMTIME) / 16 -> THMTIME;
IF NOT(VERBOSE) OR CHAROUT /= CUCHAROUT
THEN
CUCHAROUT; CHAROUT -> CUCHAROUT;
NL(1); PPR(IHM); NL(1); PR(THMTIME); NL(1);
-> CUCHAROUT;
CLOSE;
IF VERBOSE
THEN
NL(4);
PRSTRING('TIME: ');PR(THMTIME);PRSTRING(' SECS. ');
NL(5);
CLOSE;
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) = "COND"
  THEN
    IF HD(TL(TL(THM))) == NIL
      THEN
        [% "COND", [% "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;
GENRLIZE(INDTERM) -> INDTERM;
IF INDUCT(INDTERM)
  THEN
    -> INDTERM;
  ELSE INDTERM; 0; EXIT;
IF CONJFLAG
  THEN
    [% "COND", INDTERM,
      APPSUBST(GENSKOLISTT(INDVARS), 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"));
LOOP:            (REPORT("/", IDENTFN, "PROVE1"));
THM -> OLDTHM;
EVALUATE(THM) -> THM;  (REPORTIF("E", 'EVALUATION YIELDS:', "PROVE1", LASTREPTHM)
);
NORMALIZE(THM) -> THM;  (REPORTIF("N", 'WHICH NORMALIZES TO:', "PROVE1", LASTREPTHM)
);
M);

```

```

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;
  APPLIST(GETTHM(THM),
    LAMBDA THM;
    IF AVOIDSTAR AND MEMBER(" ", HD(THM)) THEN EXIT;
    PROVE1(THM); TOTTIME+THMTIME->TOTTIME; END);
  POPITON();
  NL(10);
  PRSTRING('TOTAL TIME: '); PR(TOTTIME); PRSTRING(' SECS. ');
  NL(5);
  ELSEIF HD(THM) = "T"
  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 AVUIDSTAR 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;

```

[/ IND1] TRACK 22
CREATED 11.06 25 7 1973

[21.2 14 SEPT 1973]

VARS RIDCARCDR FAILURES BOMBAY ARGLIST SCORE DESTCAND CONSCAND;

RECORDFNS("CANDREC",[0 0 0 0])

-> FAILURES

-> BOMBAY

-> ARGLIST

-> SCORE

-> DESTCAND

-> CONSCAND;

FUNCTION CARCDRSKO TERM;

LOOPIF ISLINK(TERM)

THEN

IF HD(TERM) /= "CAR" AND HD(TERM) /= "CDR"

THEN 0; EXIT;

HD(TL(TERM)) -> TERM;

CLOSE;

IF TERM = NIL OR TERM = 0 THEN 0; EXIT;

TERM;

1;

END;

FUNCTION COLLARGS BOMBLIST;

VARS ARGS BOMBS POCKET CONST;

NIL -> ARGS;

NIL -> BOMBS;

LOOPIF BOMBLIST /= NIL

THEN

HD(BOMBLIST) -> POCKET;

TL(BOMBLIST) -> BOMBLIST;

POCKET <> BOMBS -> BOMBS;

LOOPIF POCKET /= NIL

THEN

IF CARCDRSKO(HD(POCKET))

THEN

-> CONST;

IF NOT(MEMBER(CONST,ARGS))

THEN CONST::ARGS->ARGS; CLOSE;

ELSE 0; EXIT;

TL(POCKET) -> POCKET;

CLOSE;

CLOSE;

BOMBS;

ARGS;

1;

END;

FUNCTION GETCANDS ANALYSIS;

VARS ARGS BOMBS CANDLIST;

NIL -> CANDLIST;

APPLIST(ANALYSIS,

LAMBDA FAULTDESC;

IF COLLARGS(HD(TL(TL(FAULTDESC))))

```

THEN
-> ARGS -> BOMBS;
CONSCAND(1,ARGS,BOMBS,HD(TL(TL(TL(FAULTDESC))))))
:: CANDLIST -> CANDLIST;
CLOSE;
END);
CANDLIST;
END;

```

```

FUNCTION MERGECANDS CANDLIST;
VARS CAND1;
CANDLIST;
LOOPIF TL(CANDLIST) /= NIL
THEN
HD(CANDLIST) -> CAND1;
TL(CANDLIST) -> CANDLIST;
XAPPLIST(CANDLIST,
LAMBDA CAND2;
IF INTSECTP(ARGLIST(CAND1),ARGLIST(CAND2),NONOP =)
THEN
1 -> XAPPFLAG;
0 -> SCORE(CAND1);
UNION(ARGLIST(CAND1),ARGLIST(CAND2),NONOP =)->ARGLIST(CAND2);
UNION(BOMBAY(CAND1),BOMBAY(CAND2),EQUAL) -> BOMBAY(CAND2);
UNION(FAILURES(CAND1),FAILURES(CAND2),EQUAL) -> FAILURES(CAND2);
SCORE(CAND2)+1->SCORE(CAND2);
CLOSE;
END);
CLOSE;
END;

```

```

FUNCTION CHOOSEHIGH CANDLIST;
VARS HIGH ANS;
-10000 -> HIGH;
LOOPIF CANDLIST /= NIL
THEN
IF SCORE(HD(CANDLIST)) > HIGH AND SCORE(HD(CANDLIST))
THEN
SCORE(HD(CANDLIST)) -> HIGH;
HD(CANDLIST) :: NIL -> ANS;
ELSEIF SCORE(HD(CANDLIST)) = HIGH
THEN
HD(CANDLIST) :: ANS -> ANS;
CLOSE;
TL(CANDLIST) -> CANDLIST;
CLOSE;
ANS;
END;
[[CONS RAND1 RAND2] [CONS RAND3 RAND4] [CONS RAND5 RAND6]
 [CONS RAND7 RAND8] [CONS RAND9 RAND10] [CONS RAND11 RAND12]]
-> RANDCONS;

```

```

FUNCTION RATECANDS CANDLIST INDTERM;
VARS X;
APPLIST(CANDLIST,
LAMBDA CAND;
RANDCONS -> X;
ERASE(EVALUATE(APPSUBST(MAPLIST(ARGLIST(CAND)),

```

```
LAMBDA TERM;  
CONSPAIR(TERM,HD(X),TL(X)->X);  
END), INDTERM));  
STEPCNT -> SCORE(CAND);  
END);  
CANDLIST;  
END;
```

```
FUNCTION CHOOSENEW CANDLIST;  
APPLIST(CANDLIST,  
LAMBDA CAND;  
1->SCORE(CAND);  
APPLIST(ARGLIST(CAND),  
LAMBDA TERM;  
IF NOT(MEMBER(TERM,INDVARLIST))  
THEN 1 + SCORE(CAND) -> SCORE(CAND);  
CLOSE;  
END);  
END);  
CHOOSEHIGH(CANDLIST);  
END;
```

```
CONSPAIR("CAR","CARARG"):::(CONSPAIR("CDR","CDRARG"):::NIL)  
-> RIDCARCDR;
```

```
FUNCTION PICKINDVARS INDTERM;  
VARS CANDLIST;  
ERASE(EVALUATE(APPSUBST(RIDCARCDR,INDTERM)));  
GETCANDS(ANALYSIS) -> CANDLIST;  
IF CANDLIST = NIL THEN 0; EXIT;  
MERGECANDS(CANDLIST) -> CANDLIST;  
CHOOSEHIGH(CANDLIST) -> CANDLIST;  
IF TL(CANDLIST) /= NIL  
THEN  
RATECANDS(CANDLIST,INDTERM) -> CANDLIST;  
CHOOSEHIGH(CANDLIST) -> CANDLIST;  
IF TL(CANDLIST) /= NIL  
THEN  
CHOOSENEW(CANDLIST) -> CANDLIST;  
CLOSE;  
CLOSE;  
HD(CANDLIST);  
1;  
END;
```

[/ IND2] TRACK 22
CREATED 24.15 31 5 1973

[21.21 14 SEPT 1973]

VARS CARCONSTS CANDREC JUSTCARSUBST CARSUBST CARCDRINFO;

```
FUNCTION GENCARCONST L;  
MAPLIST(L,LAMBDA CONST;CONSPAIR(CONST,  
  IF ISNUMSKO(CONST) THEN NIL; ELSE GENSKO(CONST,0)CLOSE;);END);  
END;
```

```
FUNCTION SETUPSUBST;  
VARS X Y;  
NIL -> CARCDRINFO;  
NIL -> CARSUBST;  
NIL -> JUSTCARSUBST;  
APPLIST(BOMBAY(CANDREC),  
  LAMBDA TERM;  
    HD(TERM) -> X;  
    HD(TL(TERM)) -> TERM;  
    IF ASSOC(TERM,CARCDRINFO)  
      THEN  
        -> Y;  
        IF X /= BACK(Y)  
          THEN "BOTH" -> BACK(Y); CLOSE;  
        ELSE  
          CONSPAIR(TERM,X) :: CARCDRINFO -> CARCDRINFO;  
          CLOSE;  
      END);  
APPLIST(CARCDRINFO,  
  LAMBDA X;  
    IF BACK(X) /= "CDR"  
      THEN  
        CONSPAIR(FRONT(X),BACK(ERASE(ASSOC(FRONT(X),CARCONSTS)))) -> Y;  
        IF BACK(X) = "CAR"  
          THEN Y :: JUSTCARSUBST -> JUSTCARSUBST; CLOSE;  
        Y :: CARSUBST -> CARSUBST;  
        CLOSE;  
      END);  
END;
```

```
FUNCTION CONJOIN L;  
IF TL(L) = NIL  
  THEN HD(L);  
ELSE [% "AND", HD(L), CONJOIN(TL(L)) %]; CLOSE;  
END;
```

```
FUNCTION NILCASE;  
CONJOIN(MAPLIST(INDVARS,LAMBDA X;SUBST(NIL,X,INDTERM);END));  
END;
```

```
FUNCTION INDHYP;  
IF CARSUBST /= NIL
```

```

THEN
IF LENGTH(CARSUBST) = LENGTH(JUSTCARSUBST)
  THEN APPSUBST(CARSUBST,INDTERM);
  ELSE [% "AND", APPSUBST(CARSUBST,INDTERM),
        APPSUBST(JUSTCARSUBST,INDTERM) %]; CLOSE;
ELSE INDTERM; CLOSE;
END;

```

```

FUNCTION INDCONCL;
APPSUBST(MAPLIST(INDVARS,
  LAMBDA VAR;
  CONSPAIR(VAR,[% "CONS", BACK(ERASE(ASSOC(VAR,CARCONSTS))), VAR %]);
  END), INDTERM);
END;

```

```

FUNCTION SIMPLEIND;
VARS X;
BOMBAY(CANDREC) -> X;
LOOPIF X /= NIL
  THEN
  IF ISLINK(HD(TL(HD(X)))) THEN 0; EXIT;
  TL(X) -> X;
  CLOSE;
FAILURES(CANDREC) -> X;
LOOPIF X /= NIL
  THEN
  IF ISLINK(HD(TL(HD(X))))
  THEN
  IF CARCDRSKO(HD(TL(HD(X)))) AND MEMBER((),INDVARS)
  THEN 0; EXIT;
  CLOSE;
  TL(X) -> X;
  CLOSE;
1;
END;

```

```

FUNCTION SPECIAL1;
VARS X;
IF TL(INDVARS) = NIL
  THEN
  IF GENMEM([% "CDR", CONST %],BOMBLIST,
    LAMBDA;NOT(EQUAL());END)
  THEN 0; EXIT;
  FAILURES(CANDREC) -> X;
  LOOPIF X /= NIL
  THEN
  IF ISLINK(HD(TL(HD(X)))) AND CARCDRSKO(HD(TL(HD(X)))) AND
    () = CONST AND HD(HD(TL(HD(X)))) = "CAR" OR
    ISLINK(HD(TL(HD(TL(HD(X))))))
  THEN 0; EXIT;
  TL(X) -> X;
  CLOSE;
  1;
  ELSE 0; CLOSE;
END;

```

```

FUNCTION SPECIAL2;
IF TL(INDVARS) = NIL
  THEN

```



```

IF MEMBERID([%"CDR",[%"CDR",CONST%]%),BOMBLIST)
  OR MEMBERID([%"CAR",[%"CDR",CONST%]%),BOMBLIST)
  THEN 1; ELSE 0; CLOSE;
ELSE 0; CLOSE;
END;

```

```

FUNCTION SPEC2HYP;
CONJOIN([%
  IF MEMBERID([%"CAR",CONST%),BOMBLIST)
    THEN SUBST(CARCON1,CONST,INDTERM); CLOSE,
  IF MEMBERID([%"CAR",[%"CDR",CONST%] %),BOMBLIST)
    THEN SUBST(CARCON2,CONST,INDTERM);CLOSE,
  IF MEMBERID([%"CDR",[%"CDR",CONST%]%),BOMBLIST)
    THEN INDTERM; CLOSE %]);
END;

```

```

FUNCTION SPECIALMODE;
VARS CARCON1 CARCON2 CONST BOMBLIST;
HD(INDVARS) -> CONST;
BOMBBAY(CANDREC) -> BOMBLIST;
BACK(ERASE(ASSOC(CONST,CARCONSTS))) -> CARCON1;
BACK(HD(GENCARCONST([% CONST %]))) -> CARCON2;
IF SPECIAL1()
  THEN
  (REPORT("S1",'(SPECIAL CASE REQUIRED)',"SPECIALMODE"));
  [% "AND", NILCASE(),
    [% "AND", SUBST([%"CONS",CARCON1,NIL%),CONST,INDTERM),
      [% "IMPLIES", SUBST([%"CONS",CARCON2,CONST %],
        CONST,INDTERM),
        SUBST([%"CONS",CARCON1,[%"CONS",CARCON2,CONST%] %],
          CONST,INDTERM) %] %] %];

```

```

ELSEIF SPECIAL2()
  THEN
  (REPORT("S2",'(SPECIAL CASE REQUIRED)',"SPECIALMODE"));
  [% "AND", NILCASE(),
    [% "AND", SUBST([%"CONS",CARCON1,NIL%),CONST,INDTERM),
      [% "IMPLIES", SPEC2HYP(),
        SUBST([%"CONS",CARCON1,[%"CONS",CARCON2,CONST%] %],
          CONST,INDTERM) %] %] %];

```

```

ELSE
ERRFUN('SPECIAL CASE NOT COVERED',10000);
SETPOP();
CLOSE;
END;

```

```

FUNCTION INDREPORT;
IF VERBOSE
  THEN
  POPITON();
  PRSEQUEN('INDUCT ON ',INDVARS,PR);NL(2);
  CLOSE;
END;

```

```

FUNCTION INDUCT INDTERM;
(REPORT(NIL,'MUST TRY INDUCTION.','INDUCT'));
IF NOT(PICKINDVARS(INDTERM)) THEN 0; EXIT;
-> CANDREC;

```

```
ARGLIST(CANDREC) -> INDVARS;  
INDVARS <> INDVARLIST -> INDVARLIST;  
GENCARCONST(INDVARS) -> CARCONSTS;  
IF SIMPLEIND()  
  THEN  
    SETUPSUBST();  
    [% "AND", NILCASE(), [% "IMPLIES", INDHYP(), INDCONCL() %] %];  
    ELSE SPECIALMODE(); CLOSE;  
-> INDFORM;  
(REPORT(INDVARS,INDREPORT,"INDUCT"));  
INDFORM;  
1;  
END;
```

[/ VERB] TRACK 22
CREATED 17.16 9 5 1973

[21.21 14 SEPT 1973]

```
FUNCTION REPORTF1;  
IF [SLINK(CODE) THEN NL(2);PPR(INDTERM);NL(2);CLOSE;  
END;  
  
OPERATION 1 TALK; 1 -> VERBOSE; END;  
  
OPERATION 1 QUIET; 0 -> VERBOSE; IDENTFN -> REPORTFN; END;  
  
OPERATION 1 WHISPER; 0 -> VERBOSE; REPORTF1 -> REPORTFN; END;  
  
REPORTFN -> OLDREPFN;
```

[/ FERTILIZ] TRACK 22
CREATED 19.49 17 4 1973

[21.21 14 SEPT 1973]

COMMENT 'THIS IS THE FERTILIZATION FUNCTION. AGAIN, IN-LINE COMMENTS
EXPLAIN ITS BEHAVIOUR.';

VARS FERTCODE FERTLIST;

FUNCTION FERTREPORT;

IF VERBOSE

THEN

POPTON();

NL(4);PRSTRING('FERTILIZE WITH ');PRIND(TERM1,15,1);

PRSTRING('.');NL(2);

CLOSE;

END;

FUNCTION FERTILIZE TERM;

VARS TERM1 TERM2 TERM3 LHS1 LHS2 RHS1 RHS2 X;

IF SHD(TERM) /= "COND" THEN 0; EXIT;

HD(TL(TERM)) -> TERM1;

HD(TL(TL(TERM))) -> TERM2;

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

COMMENT 'FERTILIZE ONLY TERMS OF FORM (COND (EQUAL A B) C T)';

IF SHD(TERM1) = "EQUAL" AND NOT(TERM3 == NIL) AND BOOLEAN(TERM2)

AND BOOLEAN(TERM3)

THEN

HD(TL(TERM1)) -> LHS1;

HD(TL(TL(TERM1))) -> RHS1;

IF ISREALLINK(LHS1) OR ISREALLINK(RHS1)

THEN GOTO NOFERT; CLOSE;

COMMENT '(COND (EQ A B) (EQ C D) 1) => (COND (EQ C D)X 1 FU)

WHERE (EQ C D)X HAS BEEN CROSS FERTILIZED FROM (EQ A B)';

IF SHD(TERM2) = "EQUAL"

THEN

HD(TL(TERM2)) -> LHS2;

HD(TL(TL(TERM2))) -> RHS2;

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 NOXFERT; CLOSE;
CLOSE;
[% "EQUAL", LHS2, RHS2 %] -> TERM2;
"X" -> FERTCODE;

COMMENT '(COND (EQ A B) (NON-EQ) 1) => (COND (NON-EQ)A/B 1 FU)';

ELSEIF (NOXFERT: "F" -> FERTCODE; OCCUR(LHS1,TERM2))
THEN
  IF OCCUR(RHS1,TERM2)
  THEN
    IF CONSCNT(RHS1) < CONSCNT(LHS1)
    THEN SUBST(RHS1,LHS1,TERM2) -> TERM2;
    ELSE SUBST(LHS1,RHS1,TERM2) -> TERM2; CLOSE;
  ELSE SUBST(RHS1,LHS1,TERM2) -> TERM2; CLOSE;
  ELSE
  IF OCCUR(RHS1,TERM2)
  THEN SUBST(LHS1,RHS1,TERM2) -> TERM2;
  ELSE GOTO NOFERT; CLOSE;
  CLOSE;
GENSYM("*",0) -> X;
[% "COND", TERM1, NIL, TERM3 %] -> PROP("AUXDEFN",X);
1 -> PROP("BOOLEAN",X);
X :: FERTLIST -> FERTLIST;
[% "COND", TERM2, "T", [% X %] %] -> TERM2;
  (REPORT(FERTCODE,FERTREPORT,"FERTILIZE"));
IF TERM3 == 1
  THEN TERM2;
  ELSE [% "COND", TERM2, [% "COND", TERM3, "T", TERM1 %], NIL %]; CLOSE;
1;
EXIT;

COMMENT 'IF NO FERTILIZATION WAS POSSIBLE, RECURSE INTO
COMPONENTS FOR FIRST POSSIBLE ONE';

NOFERT:
1 -> TERM3;
[% "COND", APPLIST(TL(TERM),
  LAMBDA TERM;
  IF TERM3 AND FERTILIZE(TERM) THEN 0 -> TERM3; ELSE TERM; CLOSE;
  END), (IF TERM3 THEN ERASE(ERASE()),ERASE(),ERASE(),ERASE()); 0; EXIT)%];
1;

END;

```

[/ REDUCE] TRACK 22
CREATED 19.48 17 4 1973

[21.22 14 SEPT 1973]

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

VARS REDUCE;

FUNCTION REDUCE1 TERM CONSLIST;

VARS TERM1 TERM2 TERM3;

RECURSE:

COMMENT 'IF TERM IS ATOM OR NON-COND, QUIT';

IF ATOM(TERM) OR HD(TERM) /= "COND"

THEN

TERM;

EXIT;

COMMENT 'GET COMPONENTS OF THE COND';

HD(TL(TERM)) -> TERM1;

HD(TL(TL(TERM))) -> TERM2;

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

COMMENT 'IF TERM1 IS NIL OR CONS, EVAL IT';

IF TERM1 == NIL

THEN

TERM3 -> TERM;

GOTO RECURSE;

ELSEIF ISCONS(TERM1) OR MEMBERID(TERM1,CONSLIST)

THEN

TERM2 -> TERM;

GOTO RECURSE;

CLOSE;

COMMENT '(COND ATOM A B) => (COND ATOM R(A(ATOM/CONS)) R(B(ATOM/NIL)))';

IF ATOM(TERM1)

THEN

GOTO SUBSTCONS;

CLOSE;

COMMENT '(COND (EQUAL A KNOWNLINK) B C) => (COND (EQUAL A KNOWNLINK)

R(B(A/KNOWNLINK))

R(C((EQUAL A KNOWNLINK)/NIL)))';

IF HD(TERM1) = "EQUAL"

THEN

IF ISREALLINK(HD(TL(TERM1)))

THEN SUBST(HD(TL(TERM1)),HD(TL(TL(TERM1))),TERM2) -> TERM2;

ELSEIF ISREALLINK(HD(TL(TL(TERM1))))

THEN SUBST(HD(TL(TERM1)),HD(TL(TERM1)),TERM2) -> TERM2;

ELSE GOTO SUBSTRUE; CLOSE;

GOTO ASSEMBOOL;

CLOSE;

COMMENT '(COND (COND ...) A B) => (COND R(COND) R(A) R(B))';

IF HD(TERM1) = "COND"

THEN

```

REDUCE1(TERM1,CONSLIST) -> TERM1;
REDUCE1(TERM2,CONSLIST) -> TERM2;
REDUCE1(TERM3,CONSLIST) -> TERM3;
IF TERM3 == NIL THEN GOTO CONTINUE; CLOSE;
[% "COND", TERM1, TERM2, TERM3 %];
EXIT;

CONTINUE:

COMMENT '(COND BOOL A B) => (COND BOOL R(A(BOOL/T)) R(B(BOOL/NIL)))';
IF BOOLEAN(TERM1)
  THEN
SUBSTRUE:
  SUBST("T",TERM1,TERM2) -> TERM2;
ASSEMBOOL:
  [% "COND", TERM1,
    REDUCE1(TERM2,CONSLIST),
    REDUCE1(SUBST("NIL",TERM1,TERM3),CONSLIST) %];
  EXIT;

COMMENT '(COND RANDOM A B) => (COND RANDOM R(A(RANDOM/CONS))
                                R(B(RANDOM/NIL)))';
SUBSTCONS:
  [% "COND", TERM1, REDUCE1(TERM2,TERM1 :: CONSLIST),
    REDUCE1(SUBST(NIL,TERM1,TERM3),CONSLIST) %];

END;

REDUCE1(% NIL %) -> REDUCE;

```

[/ GENRLIZE] TRACK 22
CREATED 17.07 5 3 1973

[21.22 14 SEPT 1973]

COMMENT 'THIS FILE GENERALIZES THE TERM ABOUT TO BE PROVED BY INDUCTION.
WE GENERALIZE ON THE COMMON SUBTERMS ON EITHER SIDE
OF "EQUAL" AND "IMPLIES" STMTS, AND QUALIFY THE GENERALIZATIONS
WITH TYPE STATEMENTS.';

COMMENT 'FIND ALL COMMON NON-ATOMIC NON-PRIMITIVE SUBTERMS OF TWO TERMS.';

VARS T2 GENRLTLIST ATOMLIST;

```
FUNCTION COMSUBT1 T1;
VARS X;
IF ATOM(T1)
  THEN
    OCCUR(T1,T2);
  ELSE
    TL(T1) -> X;
    IF (1;LOOPIF X/=NIL THEN LOGAND(COMSUBT1(HD(X)));TL(X)->X;CLOSE;)
      THEN
        IF NOT(LISPPRIM(T1)) AND OCCUR(T1,T2)
          THEN
            IF NOT(MEMBERID(T1,GENRLTLIST))
              THEN T1 :: GENRLTLIST -> GENRLTLIST;CLOSE;
            1; EXIT;
          CLOSE;
        0;
      CLOSE;
END;
```

```
FUNCTION COMSUBTERMS T1 T2;
IF CONSCNT(T1) > CONSCNT(T2) THEN T1;T2->T1->T2;CLOSE;
ERASE(COMSUBT1(T1));
END;
```

COMMENT 'FIND ALL COMMON SUBTERMS OCCURRING ACROSS EQS AND
IMPLIES.';

```
FUNCTION GENRLT1 TERM;
IF ATOM(TERM) THEN EXIT;
IF HD(TERM) = "EQUAL"
  THEN
    COMSUBTERMS(HD(TL(TERM)),HD(TL(TL(TERM))));
  ELSEIF HD(TERM) = "COND" AND HD(TL(TL(TL(TERM)))) == 1
    THEN
      IF ATOM(HD(TL(TERM)))
        THEN;
      ELSE APPLIST(TL(HD(TL(TERM))),LAMBDA;COMSUBTERMS(HD(TL(TL(TERM))));
        END); CLOSE;
```



```
CLOSE;
APPLIST(TL(TERM),GENRLT1);
END;
```

```
FUNCTION GENRLTERMS;
VARS GENRLTLIST;
NIL -> GENRLTLIST;
GENRLT1();
GENRLTLIST;
END;
```

```
COMMENT 'QUALIFY THE GENERALIZATION BY ADDING TYPE STMTS';
```

```
FUNCTION ADDTYPESTMTS LIST TERM;
VARS X;
IF LIST = NIL
  THEN TERM;
ELSE
  IF OCCUR(BACK(HD(LIST)),TERM)
    THEN
      TYPEEXPR(FRONT(HD(LIST))) -> X;
      IF HD(X) /= "CONSTTRUE"
        THEN
          [% "IMPLIES", SUBST(BACK(HD(LIST)),"X",X),
            ADDTYPESTMTS(TL(LIST),TERM) %]
        ELSE
          ADDTYPESTMTS(TL(LIST),TERM);
          CLOSE;
      ELSE
        ADDTYPESTMTS(TL(LIST),TERM);
        CLOSE;
        CLOSE;
END;
```

```
COMMENT 'THIS FUNCTION MAKES A VERBOSE REPORT ON THE PROGRESS
OF GENERALIZATION.';
```

```
FUNCTION GENREPORT;
IF VERBOSE
  THEN
    POPTON();
    NL(2);
    PRSEQUEN('GENERALIZE COMMON SUBTERMS BY REPLACING ',
      SUBSTLIST, LAMBDA P;PR(FRONT(P));PRSTRING(' BY ');PR(BACK(P));END);
    NL(2);
    PRSTRING('THE GENERALIZED TERM IS:');
    NL(2);
    PPR(TERM);
    NL(2);
    CLOSE;
END;
```

```
COMMENT 'THIS IS THE TOP-LEVEL FUNCTION. IT GENERALIZES ITS
ARGUMENT AS DESCRIBED, QUALIFIES IT, AND THEN PRINTS
```

```
A VERBOSE COMMENT IF NEEDED.';

FUNCTION GENRLIZE TERM;
VARS X SUBSTLIST;
GENRLTERMS(TERM) -> X;
IF X = NIL THEN TERM; EXIT;
MAPLIST(X,
  LAMBDA T;
    GENSKO("GENRL",0) -> X;
    CONSPAIR(T,X);
  END) -> SUBSTLIST;
IF SUBSTLIST = NIL THEN TERM; EXIT;
SUBSTLIST <> GENRLLIST -> GENRLLIST;
ADDTYPESTMTS(SUBSTLIST,APPSUBST(SUBSTLIST,TERM))
-> TERM;
  (REPORT("G",GENREPORT,"GENRLIZE"));
TERM;
END;
```