Alan

```
LL          MM      MM
LL          MM      MM
LL          MMMM  MMMM
LL          MMMM  MMMM
LL          MM  MM  MM
LL          MM  MM  MM
LL          MM      MM
LL          MM      MM
LL          MM      MM
LL          MM      MM
LL          MM      MM      ....
LL          MM      MM      ....
LLLLLLLLLL  MM      MM      ....
LLLLLLLLLL  MM      MM      ....


44    44      ØØØØØØ      ØØØØØØ                44    44        11        44    44
44    44      ØØØØØØ      ØØØØØØ                44    44        11        44    44
44    44    ØØ      ØØ  ØØ      ØØ              44    44      1111        44    44
44    44    ØØ      ØØ  ØØ      ØØ              44    44      1111        44    44
44    44    ØØ    ØØØØ  ØØ    ØØØØ              44    44        11        44    44
44    44    ØØ    ØØØØ  ØØ    ØØØØ              44    44        11        44    44
4444444444  ØØ  ØØ  ØØ  ØØ  ØØ  ØØ              4444444444      11        4444444444
4444444444  ØØ  ØØ  ØØ  ØØ  ØØ  ØØ              4444444444      11        4444444444
      44    ØØØØ    ØØ  ØØØØ    ØØ                    44        11              44
      44    ØØØØ    ØØ  ØØØØ    ØØ                    44        11              44
      44    ØØ      ØØ  ØØ      ØØ      ,,,,          44        11              44
      44    ØØ      ØØ  ØØ      ØØ      ,,,,          44        11              44
      44      ØØØØØØ      ØØØØØØ          ,,          44      111111            44
      44      ØØØØØØ      ØØØØØØ          ,,          44      111111            44


W   W EEEEE L     H   H  AAA  M   M       BBBB
W   W E     L     H   H A   A MM MM       B   B
W   W E     L     H   H A   A M M M       B   B
W   W EEEE  L     HHHHH A   A M   M       BBBB
W W W E     L     H   H AAAAA M   M       B   B
WW WW E     L     H   H A   A M   M       B   B
W   W EEEEE LLLLL H   H A   A M   M       BBBB


LPTSPL VERSION 6(344)  RUNNING ON LPT5ØØ
 *START* USER WELHAM B [4ØØ,414] JOB  LM SEQ. 3238 DATE 17-NOV-75 15:Ø7:46 MONITOR NETMON 5.Ø7B V3 (AUG 12 T *START*
REQUEST CREATED: 17-NOV-75  15:Ø7:3Ø
FILE: DSKA1:LM[4ØØ,414] CREATED: 17-NOV-75 15:Ø2:ØØ <155>  PRINTED: 17-NOV-75 15:Ø7:56
QUEUE SWITCHES:  /PRINT:ARROW /FILE:ASCII /COPIES:2 /SPACING:1 /LIMIT:5Ø /FORMS:NORMAL
FILE WILL BE RENAMED TO <Ø55> PROTECTION
```

```
* LEGAL MOVES FOR PROLOG EQUATION SOLVING PROGRAM.
* BOB WELHAM                     1975.


+&(DG,2).            * INFIX CONJUNCTION.
+#(DG,1).            * INFIX DISJUNCTION.
+=(DG,3).            * EQUATION SYMBOL.


* ARITHMETIC OPERATORS.

+:(GD,8).            * EXPONENTIATION.
++(GD,4).            * ADDITION.
+-(GD,9).            * UNARY MINUS.
+.(DG,6).            * MULTIPLICATION.




****************************************.
* LEGAL MOVES.
****************************************.

* LEGAL MOVES FOR SIMPLIFICATION.

+AX1( *U+Ø , *U , TRUE ).

+AX1Ø( *U.Ø , Ø , TRUE ).

+AX2Ø( *U:Ø , 1 , TRUE ).

+AX3Ø( Ø:*U , Ø , NONZERO(*U) ).

+AX4Ø( LOG(*U,1) , Ø , TRUE ).

+AX5Ø( LOG(*U,*U) , 1 , TRUE ).

+AX6Ø( *U.1 , *U , TRUE ).

+AX8Ø( *U:1 , *U , TRUE ).

+AX85( 1:*U , 1 , TRUE ).

+AX9Ø( *U+-1.*U , Ø , TRUE ).

+AX91( 1+-1 , Ø , TRUE ).

+AX1ØØ( -1.-1 , 1 , TRUE ).

+AX1Ø1( -1:-1 , -1 , TRUE ).

+AX2ØØ( LOG(*U,*U:*V) , *V , TRUE ).

+AX2Ø1( *U:LOG(*U,*V) , *V , TRUE ).
```

```
* LEGAL MOVES FOR ISOLATION.

+AX1000( *U+*V=*W , *U=*W+-1.*V , TRUE ).

+AX1020( *U.*V=*W , *U=*W.*V:-1 , NONZERO(*V) ).

+AX1040( LOG(*U,*V)=*W , *V=*U:*W , TRUE ).

+AX1050( *U:*N=*V , *U=*V:(*N:-1) #
                    *U=-1.*V:(*N:-1) , EVEN(*N) ).

+AX1060( *U:*N=*V , *U=*V:(*N:-1) , ODD(*N) ).

+AX1070( *U:*V=*W , *V=LOG(*U,*W) , TRUE ).

+AX1500( SIN(*U)=*V , *U=*N.PI+-1:*N.ARCSIN(*V) , ARBINT(*N) ).

+AX1510( COS(*U)=*V , *U=2.*N.PI+ARCCOS(*V) #
                      *U=2.*N.PI+-1.ARCCOS(*V) , ARBINT(*N) ).

+AX1520( TAN(*U)=*V , *U=*N.PI+ARCTAN(*V) , ARBINT(*N) ).

+AX1530( COSEC(*U)=*V , *U=*N.PI+-1:*N.ARCCOSEC(*V) , ARBINT(*N) ).

+AX1540( SEC(*U)=*V , *U=2.*N.PI+ARCSEC(*V) #
                      *U=2.*N.PI+-1.ARCSEC(*V) , ARBINT(*N) ).

+AX1550( COT(*U)=*V , *U=*N.PI+ARCCOT(*V) , ARBINT(*N) ).

+AX1560( ARCSIN(*U)=*V , *U=SIN(*V) , TRUE ).

+AX1570( ARCCOS(*U)=*V , *U=COS(*V) , TRUE ).

+AX1580( ARCTAN(*U)=*V , *U=TAN(*V) , TRUE ).

+AX1590( ARCCOSEC(*U)=*V , *U=COSEC(*V) , TRUE ).

+AX1600( ARCSEC(*U)=*V , *U=SEC(*V) , TRUE ).

+AX1610( ARCCOT(*U)=*V , *U=COT(*V) , TRUE ).



* LEGAL MOVES FOR COLLECTION.

+AX2000( (*U+*V).*W , *U.*W+*V.*W , TRUE ).

+AX2001( (*V+1).*W , *W+*V.*W , TRUE ).

+AX2002( 2.*W , *W+*W , TRUE ).

+AX2010( (*U+*V).(*U+-1.*V) , *U:2+-1.*V:2 , TRUE ).

+AX2011( (*U+1).(*U+-1) , *U:2+-1 , TRUE ).

+AX2020( *W:(*U+*V) , *W:*U.*W:*V , TRUE ).
```

```
+AX2021( *W:(*V+1) , *W.*W:*V , TRUE ).

+AX2022( *W:2 , *W.*W , TRUE ).

+AX2500( SIN(2.*U).2:-1 , SIN(*U).COS(*U) , TRUE ).

+AX2510( COS(2.*U) , COS(*U):2+-1.SIN(*U):2 , TRUE ).

+AX2520( SIN(*U+*V) , SIN(*U).COS(*V)+COS(*U).SIN(*V) , TRUE ).

+AX2530( SIN(*U+-1.*V) , SIN(*U).COS(*V)+-1.COS(*U).SIN(*V) , TRUE ).

+AX2540( COS(*U+*V) , COS(*U).COS(*V)+-1.SIN(*U).SIN(*V) , TRUE ).

+AX2550( COS(*U+-1.*V) , COS(*U).COS(*V)+SIN(*U).SIN(*V) , TRUE ).




* LEGAL MOVES FOR ATTRACTION.

+AX3000( LOG(*W,*U)+LOG(*W,*V) , LOG(*W,*U.*V) , TRUE ).

+AX3001( LOG(*W,*U)+*A.LOG(*W,*V) , LOG(*W,*U.*V:*A) , TRUE ).

+AX3010( (*U:*V):*W , *U:(*V.*W) , TRUE ).




*********************************************.
* RECOMMENDATION LISTS.
*********************************************.

* ALL AXIOMS APPLIED LEFT TO RIGHT FOR ISOLATION
  AND SIMPLIFICATION.

+ISOLATES(1000).
+ISOLATES(1020).
+ISOLATES(1040).
+ISOLATES(1050).
+ISOLATES(1060).
+ISOLATES(1070).
+ISOLATES(1500).
+ISOLATES(1510).
+ISOLATES(1520).
+ISOLATES(1530).
+ISOLATES(1540).
+ISOLATES(1550).
+ISOLATES(1560).
+ISOLATES(1570).
+ISOLATES(1580).
+ISOLATES(1590).
```

```
+ISOLATES(1600).
+ISOLATES(1610).


+SIMPLIFIES(1).
+SIMPLIFIES(10).
+SIMPLIFIES(20).
+SIMPLIFIES(30).
+SIMPLIFIES(40).
+SIMPLIFIES(50).
+SIMPLIFIES(60).
+SIMPLIFIES(70).
+SIMPLIFIES(80).
+SIMPLIFIES(85).
+SIMPLIFIES(90).
+SIMPLIFIES(91).
+SIMPLIFIES(65).
+SIMPLIFIES(100).
+SIMPLIFIES(101).
+SIMPLIFIES(110).
+SIMPLIFIES(200).
+SIMPLIFIES(201).


+COLLECTS(2000:RTL).
+COLLECTS(2001:RTL).
+COLLECTS(2002:RTL).
+COLLECTS(2020:RTL).
+COLLECTS(2021:RTL).
+COLLECTS(2022:RTL).
+COLLECTS(2010:LTR).
+COLLECTS(2011:LTR).
+COLLECTS(2500:RTL).
+COLLECTS(2510:RTL).
+COLLECTS(2520:RTL).
+COLLECTS(2530:RTL).
+COLLECTS(2540:RTL).
+COLLECTS(2550:RTL).


+ATTRACTS(2000:RTL).
+ATTRACTS(2020:RTL).
+ATTRACTS(3000:RTL).
+ATTRACTS(3001:RTL).
+ATTRACTS(3010:RTL).




*********************************************.
* ARITHMETIC.
*********************************************.

+INTEGER(-1).
+INTEGER(-1.*N)-/-NATNUM(*N).
+INTEGER(*N)-NATNUM(*N).
```

```
+NATNUM(*N)-UNIV(*N,*M.NIL)-DIGITS(*M).
+DIGITS(NIL)-/.
+DIGITS(*D.*L)-CHIFFRE(*D)-DIGITS(*L).

+ODD(*N)-NATNUM(*N)-RESTE(*N,2,*R)-IDEN(*R,1).
+EVEN(*N)-NATNUM(*N)-RESTE(*N,2,*R)-IDEN(*R,Ø).

+NONZERO(*N)-NATNUM(*N)-DIFF(*N,Ø).
+NONZERO(*X)-SIMPLIFY(*X,*Y)-IDEN(*Y,Ø)-/-FAIL.
+NONZERO(*X)-SORCHA("ASSUMING NON-ZERO ")-SORTER(*X)-LIGNE.



* EVALUATE AN ARITHMETIC EXPRESSION.

+EVAL(-1.-1.*N,*R)-/-EVAL(*N,*R).
+EVAL(-1,-1.1)-/.
+EVAL(*L+*M,*N)-/-EVAL(*L,*P)-EVAL(*M,*Q)-ADD(*P,*Q,*N).
+EVAL(*L.*M,*N)-/-EVAL(*L,*P)-EVAL(*M,*Q)-TIMES(*P,*Q,*N).
+EVAL(*L:*M,*N)-/-EVAL(*L,*P)-EVAL(*M,*Q)-POWER(*P,*Q,*N).
+EVAL(*N,*N)-INTEGER(*N)-/.
+EVAL(*E,*R)-FACT(*E=*R).



* ADD,TIMES AND POWER WORK FOR PLAIN INTEGERS OR FOR INTEGERS
* PRECEDED BY ONE -1 ONLY.

+ADD(-1.*L,-1.*M,-1.*N)-/-PLUS(*L,*M,*N).
+ADD(-1.*L,*M,-1.*N)-INF(*M,*L)-/-MOINS(*L,*M,*N).
+ADD(-1.*L,*M,*N)-/-MOINS(*M,*L,*N).
+ADD(*L,-1.*M,*N)-/-ADD(-1.*M,*L,*N).
+ADD(*L,*M,*N)-PLUS(*L,*M,*N).

+TIMES(-1.*L,-1.*M,*N)-/-MULT(*L,*M,*N).
+TIMES(-1.*L,*M,-1.*N)-/-MULT(*L,*M,*N).
+TIMES(*L,-1.*M,-1.*N)-/-MULT(*L,*M,*N).
+TIMES(*L,*M,*N)-MULT(*L,*M,*N).

+POWER(-1.*L,*M,*N)-EVEN(*M)-/-POWER(*L,*M,*N).
+POWER(-1.*L,*M,-1.*N)-ODD(*M)-/-POWER(*L,*M,*N).
+POWER(*L,1,*L)-/.
+POWER(*L,*M,*N)-NATNUM(*M)-MOINS(*M,1,*P)
    -POWER(*L,*P,*Q)-MULT(*L,*Q,*N).




*********************************************.
* UTILITY ROUTINES.
*********************************************.

* COUNT NUMBER OF OCCURENCES OF GIVEN CONSTANT IN GIVEN EXPRESSION.

+OCC(*X,*X,1)-/.
```

```
+OCC(NIL,*X,0)-/.
+OCC(*E,*X,*N)-UNIV(*E,*F.*E1.*E2.NIL)-/
     -OCC(*E1,*X,*N1)-OCC(*E2,*X,*N2)-PLUS(*N1,*N2,*N).
+OCC(*E,*X,*N)-UNIV(*E,*F.*A)-OCC(*A,*X,*N).


+FREEOF(*E,*X)-OCC(*E,*X,*N)-IDEN(*N,0).
+SINGLEOCC(*E,*X)-OCC(*E,*X,*N)-IDEN(*N,1).
+CONTAINS(*E,*X)-OCC(*E,*X,*N)-DIFF(*N,0).




+SUBSTITUTE(*SUB1#*SUB2,*OLD,*NEW1#*NEW2)
     -SUBSTITUTE(*SUB1,*OLD,*NEW1)
     -SUBSTITUTE(*SUB2,*OLD,*NEW2).

+SUBSTITUTE(*U=*TERM,*OLD,*NEW)
     -TRACE(SUBSTITUTING-*TERM-FOR-*U-IN-*OLD)
     -SUB(*U=*TERM,*OLD,*NEW)-TRACE(GIVES-*NEW).


+SUB(*U=*T,*U,*T)-/.

+SUB(*U=*T,*E,*E)-FREEOF(*E,*U)-/.

+SUB(*U=*T,*E,*R)-UNIV(*E,*F.*A.*L)
     -SUB(*U=*T,*A,*B)-SUB(*U=*T,*L,*M)-UNIV(*R,*F.*B.*L).



* LIST PROCESSING.

+MEMBER(*X,*L)-APPEND(*L1,*X.*L2,*L).

+SELECT(*U,*L,*R)-APPEND(*L1,*U.*L2,*L)-APPEND(*L1,*L2,*R)
     -TRACE(*U-SELECTED-FROM-LIST-*L).

+APPEND(NIL,*L,*L).
+APPEND(*X.*L1,*L2,*X.*L)-APPEND(*L1,*L2,*L).


+SELECTA(*U,*C,*R)-APPENDA(*C1,*U&*C2,*C)-APPENDA(*C1,*C2,*R)
     -TRACE(*U-SELECTED-FROM-CONJUNCTION-*C).

+APPENDA(TRUE,*C,*C).
+APPENDA(*X&*C1,*C2,*X&*C)-APPENDA(*C1,*C2,*C).


* GENERATE IDENTIFIERS DENOTING ARBITRARY INTEGERS.

+ARBINT(*N)-AINO(*X)-UNIV(*X,*Y.NIL)-UNIV(*N,(N.*Y).NIL)
     -SUPP((+AINO(*X)).NIL)-PLUS(*X,1,*Z)
     -AJOUT((+INTEGER(*N)).NIL)-AJOUT((+AINO(*Z)).NIL)
     -SORTER(*N)-SORCHA(" DENOTES AN ARBITRARY INTEGER")-LIGNE.

+AINO(0).
```

```
+IDEN(*X,*X).

+DIFF(*X,*X)-/-FAIL.
+DIFF(*X,*Y).

+TRUE.

+PERM2(*X,*Y,*X,*Y).
+PERM2(*X,*Y,*Y,*X).




****************************************.
* USER COMMUNICATION ROUTINES.
****************************************.

+ANSWER(*ANS1#*ANS2)-/-ANSWER(*ANS1)
    -LIGNE-SORCHA("OR")-LIGNE-ANSWER(*ANS2).

+ANSWER(*ANS)-LIGNE-LIGNE
    -SORCHA("ANSWER IS ")-SORTER(*ANS)-LIGNE-LIGNE.


+TRACE(*X)-TFLAG-/-SORTER(*X)-LIGNE.
+TRACE(*X).

+T-AJOUT((+TFLAG).NIL).        * TRACE ON.
+NT-SUPP((+TFLAG).NIL).        * TRACE OFF.


+SOLVE(*EQN&*EQNS,*US)-/-SIMSOLVE(*EQN&*EQNS,*US,*ANS).
+SOLVE(*EQN,*U)-SOLVE11(*EQN,*U,*ANS).




****************************************.
* ROUTINES TO APPLY THE LEGAL MOVE AXIOMS.
****************************************.

+APPLY(LTR,*AXIOM,*OLD,*NEW)
    -UNIV(*AXIOM,*N.NIL)
    -UNIV(*GOAL,(A.X.*N).*OLDM.*NEW.*CONDITION.NIL)-*GOAL
    -MATCH(*OLD,*OLDM)-*CONDITION
    -TRACE(AXIOM-*AXIOM-LEFT-TO-RIGHT-ON-*OLD-GIVES-*NEW).

+APPLY(RTL,*AXIOM,*OLD,*NEW)
    -UNIV(*AXIOM,*N.NIL)
    -UNIV(*GOAL,(A.X.*N).*NEW.*OLDM.*CONDITION.NIL)-*GOAL
    -MATCH(*OLD,*OLDM)-*CONDITION
    -TRACE(AXIOM-*AXIOM-RIGHT-TO-LEFT-ON-*OLD-GIVES-*NEW).




****************************************.
* PATTERN MATCHER.
****************************************.

+MATCH(*X,*X).

+MATCH(*X+*Y,*U+*V)-PERM2(*X,*Y,*P,*Q)
    -MATCH(*P,*U)-MATCH(*Q,*V)-DIFF(*X+*Y,*U+*V).

+MATCH(*X.*Y,*U.*V)-PERM2(*X,*Y,*P,*Q)
    -MATCH(*P,*U)-MATCH(*Q,*V)-DIFF(*X.*Y,*U.*V).

+MATCH(*X+*Y+*Z,*Z+*X+*Y)-DIFF(*X+*Y+*Z,*Z+*X+*Y).
+MATCH(*X+*Y+*Z,*Y+*Z+*X)-DIFF(*X+*Y+*Z,*Y+*Z+*X).

+MATCH(*X.*Y.*Z,*Z.*X.*Y)-DIFF(*X.*Y.*Z,*Z.*X.*Y).
+MATCH(*X.*Y.*Z,*Y.*Z.*X)-DIFF(*X.*Y.*Z,*Y.*Z.*X).



+FIN.
```