```
    EQ1: X = (X/Y)*Y + MOD(X,Y) BY FUNCTION,DIVISION(X,Y);
    EQ2: MOD(X,Y)>=0 BY FUNCTION,MOD(X,Y);
    IEQ0: X>=(X/Y)*Y BY ARITH,EQ1,-,EQ2;
    /# SET UP CONDITIONS FOR MONOTONICITY LEMMAS                    #/

    X/Y > 0 BY FUNCTION,DIVISION(X,Y);
    (X/Y)*Y > 0 BY ARITH,X/Y>0,*,Y>0;
    /# TERMINATION OF LOG #/
    SOME D FIXED.(D>=0 & Y<=D) BY INTRO,Y;
    SOME D FIXED.(D>=0 & X/Y<=D) BY INTRO,X/Y;
    I1: LOG(B,X/Y) >= 0 BY FUNCTION,LOG(B,X/Y);
    I2: LOG(B,Y) >= 0 BY FUNCTION,LOG(B,Y);
    LOG(B,X/Y) + LOG(B,Y) >= 0 BY ARITH,I1,+,I2;
    EZ1: B**LOG(B,X/Y) > 0 BY ALLEL,EXP_POSITIVE,B,LOG(B,X/Y);
    EZ2: B**LOG(B,Y) > 0 BY ALLEL,EXP_POSITIVE,B,LOG(B,Y);
    IEQ1: B**LOG(B,X/Y) <= X/Y BY FUNCTION,LOG(B,X/Y);
    IEQ2: B**LOG(B,Y) <= Y BY FUNCTION,LOG(B,Y);
    IEQ3: B**(LOG(B,X/Y)+LOG(B,Y)) > 0
    BY ALLEL,EXP_POSITIVE,B,(LOG(B,X/Y)+LOG(B,Y));
    /# BECAUSE OF LOG PROPERTIES WE MUST RAISE LOG TO EXP           #/
    /# IN ORDER TO COMPARE ARGUMENTS.                               #/

    EQ3: B**LOG(B,X/Y)*B**LOG(B,Y) = B**(LOG(B,X/Y)+LOG(B,Y))
         BY ALLEL,EXP_ADDITIVE,B,LOG(B,X/Y),LOG(B,Y);
    /# NOW MULTIPLY IEQ1 BY IEQ2 IN STAGES                          #/
    L1: B**LOG(B,X/Y)*B**LOG(B,Y) <= (X/Y)*B**LOG(B,Y)
        BY ARITH,IEQ1,*,EZ2;
    L2: (X/Y)*B**LOG(B,Y) <= (X/Y)*Y BY ARITH,IEQ2,*,X/Y>0;
    L3: B**LOG(B,X/Y)*B**LOG(B,Y) <= (X/Y)*Y
        BY ARITH,L1,L2;
    B**(LOG(B,X/Y) + LOG(B,Y)) <= (X/Y)*Y; /# SUBST IN EQ3          #/
    /# NOW TAKE LOGARITHMS AND COMPARE                              #/
    LOG(B,B**(LOG(B,X/Y)+LOG(B,Y)))<=LOG(B,(X/Y)*Y)
    BY ALLEL,LOG_MONOTONE,B,B**(LOG(B,X/Y)+LOG(B,Y)),(X/Y)*Y;
    LOG(B,B**(LOG(B,X/Y)+LOG(B,Y))) = LOG(B,X/Y) + LOG(B,Y)
    BY ALLEL,LOG_EXACT,B,LOG(B,X/Y)+LOG(B,Y);
    F1: LOG(B,X/Y) + LOG(B,Y) <= LOG(B,(X/Y)*Y);
    F2: LOG(B,(X/Y)*Y) <= LOG(B,X)
        BY ALLEL, LOG_MONOTONE,B,(X/Y)*Y,X; /# FROM IEQ0           #/
    CONCLUSION: LOG(B,X/Y) + LOG(B,Y) <= LOG(B,X)
            BY ARITH,F1,F2;
QED;
*/

/* **** THIS IS THE FILE DIVTHY2 PLCV, ADVANCED DIVISION THEORY **** */
PRIME_FACTORIZATION: PROCEDURE(N,A,L,U,M  /*: ,AA */);
  DECLARE (A(*),M) FIXED;  /* READWRITE PARAMETERS                */
  DECLARE (N,L,U) FIXED  /*/ READONLY  */ ;
  /*: DECLARE (AA(*)) FIXED  READONLY;                             */
  /* FACTOR AN INTEGER GREATER THAN 1 INTO A PRODUCT               */
  /* OF PRIMES, PROD(A,L,M).  THE PRIMES APPEAR IN NONDECREASING   */
  /* ORDER, A(I)<=A(J) FOR I<=J.  M IS THE COMPUTED NUMBER OF FAC- */
  /* ORS WHILE U IS THE UPPER BOUND OF THE ARRAY A WHICH IS AN      */
```



```
1073    /* ESTIMATE OF THE NUMBER OF FACTORS.  LOG TO THE BASE 2 OF N IS  */
1073    /* THE ESTIMATE (CLEARLY AT LEAST THIS MANY IS REQUIRED FOR N A   */
1073    /* POWER OF 2).  THE NEED TO ESTIMATE THE NUMBER OF FACTORS IS    */
1073    /* DUE TO PL/1'S REQUIREMENT ON ARRAY DECLARATIONS.  THIS IS AN   */
1073    /* ANNOYING FEATURE OF A PL/1 BASED THEORY WHICH WOULD NOT APPEAR */
1073    /* IN, SAY, A LISP BASED THEORY.                                  */
1073    /*/ ASSUME
            DOM(A,L) & DOM(A,U) & U-L >= LOG(2,N) & N > 1, AA = A;      */
1074    /*/ ATTAIN
1074        AT1: N = PROD(A,L,M) & L<=M<=U, DOM(A,M),
1074        AT2: ALL I FIXED WHERE L<=I<=M.(PRIME(A(I)) & DIV(A(I),N) ),
1074        AT3: ALL (I,J) FIXED WHERE L<=I<=J<=M. A(I)<=A(J),
1074        AT4: ALL I FIXED WHERE LBOUND(A,1)<=I<L.AA(I)=A(I);        */
1075    /*/ ARBITRARY D FIXED WHERE U-L < D;                           */
1076        DECLARE (P,N2) FIXED;
1077        /*: DECLARE AAA(LBOUND(A,1): HBOUND(A,1)) FIXED;           */
1078        /*: DECLARE AR(LBOUND(A,1):HBOUND(A,1)) FIXED;             */
1079      /*/ SOME D FIXED.(D>=0 & N<=D) BY INTRO,N;                   */
1080      /*/ LOG(2,N)>=0 BY FUNCTION,LOG(2,N);
1081          U-L>=0 BY ARITH,U-L>=LOG(2,N)>=0;
1082          L<=U BY ARITH,U-L>=0,+,L=L;
1083          ^(U-L<0) BY ARITH,U-L>=LOG(2,N)>=0;
1084          N^=0 BY ARITH,N>1;                                        */
1085    /* FIND THE LEAST PRIME FACTOR OF N, CALL IT P                  */
1085    P = LEAST PRIME_FACTOR(N);
1086    /*/ P = LEAST PRIME FACTOR(N);                                 */
1087    /*/ PRIME(P) & DIV(P,N) & ALL I FIXED WHERE 1<I<P.^DIV(I,N)
1087        BY FUNCTION,LEAST PRIME_FACTOR(N);                        */
1088    /* DIVIDE OUT THE LEAST PRIME FACTOR AND MAKE IT THE FIRST      */
1088    /* FACTOR OF THE PRODUCT, A(L).  IF N IS COMPLETELY FACTORED    */
1088    /* AS A RESULT, THEN STOP, OTHERWISE FACTOR N/P IN THE SAME WAY. */
1088    /*/ P^=0 BY ARITH,P>1;                                         */
1089    /*/ MOD(N,P) = 0 BY ALLEL, DIV_MOD_EQUIVALENCE,N,P;            */
1090    /*/ ATTAIN N/P^=1 &  AA = A;                                   */
1091    IF N/P = 1
1091    THEN
1091       DO;
1092       /* N IS COMPLETELY FACTORED. ALL THE REQUIRED PROPERTIES     */
1092       /* AT1,AT2,AT3, CAN BE PROVED TRIVIALLY FROM THE INFORMATION */
1092       /* THAT P IS PRIME, P DIVIDES N, N/P=1, A(L)=P AND M=L.      */
1092       /* FOR THE ARRAY RULE SAVE A IN AR                           */
1092       /*: AR = A;                                                 */
1093       /*/ AR = A; AR = AA;                                        */
1095       M =L; ;
1097       /*/ M = L; A(L) = P; DOM(A,M); M<=L; M<=L<=U; DOM(A,M);
1103       /# BY ARRAY ASSIGNMENT CONCLUDE:                          #/
1103       ASGN1: ALL I FIXED WHERE I^=L.AR(I) = A(I);
1104       SOME D FIXED.(D>=0 & N<=D) BY INTRO,N;
1105       SOME D FIXED.(D>=0 & MONUS(M+1,L)<D) BY FUNCTION,
1105       PROD_TERMINATION(A,L,M);
1106        PROD(A,L,M) = A(L) BY FUNCTION,PROD(A,L,M);
1107        N = (N/P)*P + MOD(N,P) BY FUNCTION,DIVISION(N,P);
1108        N = 1*P + 0;
```

```
      N = P = PROD(A,L,M);
      M <= U;
      ALL I FIXED WHERE L<=I<=M.(PRIME(A(I))&DIV(A(I),N)) BY INTRO,
      PROOF;
      I = L BY ARITH,L<=I<=M,L=M;
      A(I) = P;
      QED;
      ALL (I,J) FIXED WHERE L<=I<=J<=M. A(I)<=A(J) BY INTRO,
      PROOF;
        I = L BY ARITH,L<=I<=J<=M,L=M;
        J = L BY ARITH,L<=I<=J<=M,L=M;
        A(I) <= A(J) BY ARITH,A(I)=A(J);
      QED;
      ALL I FIXED WHERE LBOUND(A,1)<=I<L. AA(I) = A(I) BY INTRO,
       PROOF;
          I^=L BY ARITH,I<L;
          AR(I) = A(I) BY ALLEL,ASGN1,I;                          #/
          AR(I) = AA(I); /# FROM AR = AA
       QED;
       */
       RETURN;                                                    */
      END;                                                        */
/* N IS NOT COMPLETELY FACTORED, SO FACTOR N/P IN THE SAME       */
/* WAY, BY CALLING PRIME_FACTORIZATION.  IT MUST BE SHOWN        */
/* THAT THE INPUT CONDITIONS TO PRIME_FACTORIZATION ARE MET      */
/* WE FIRST SHOW THAT THE ARRAY LENGTH IS ADEQUATE TO CONTAIN    */
/* ALL THE FACTORS.  WE SHOW THIS FIRST BECAUSE IT IS MOST IN-   */
/* TERESTING.  THE PROOF OF THE DOMAIN CONDITIONS IS MOST BORING */
/* AND IS A PRIME CANDIDATE FOR AUTOMATION IN A THEORY OF SUCC-  */
/* ESSOR.                                                        #/
      /*/
      /# CONDITIONS NEEDED TO USE LOG_OF_PROD
      N>0; P>0;
      P<=N BY ALLEL,DIVISOR_SIZE,P,N;
      2>1; 0<P<=N;
      L0: LOG(2,N/P) + LOG(2,P) <= LOG(2,N)
      BY ALLEL,LOG_OF_PRODUCT,2,N,P;
     L1: LOG(2,N/P) <= LOG(2,N)-LOG(2,P) BY ARITH,L0,-,LOG(2,P)=LOG(2,P);
     P>=2 BY ARITH,P>1;
     SOME D FIXED.(D>=0 & P<=D) BY INTRO,P;
     L2: LOG(2,P)>=1 BY ALLEL,LOG_POSITIVE,2,P;
     L3: LOG(2,N)-LOG(2,P) <= LOG(2,N)-1
     BY ARITH,LOG(2,N)=LOG(2,N),-,LOG(2,P) >= 1;
     L4: ( LOG(2,N/P) <= LOG(2,N)-1 ) BY ARITH,L1,L3;
     U-(L+1)>=LOG(2,N)-1 BY ARITH,U-L>=LOG(2,N),-,1=1;
     U-(L+1)>=LOG(2,N/P) BY ARITH,
     U-(L+1)>=LOG(2,N)-1>=LOG(2,N/P);                             #/
     /# SHOW THAT L+1 BELONGS TO THE DOMAIN (THIS DEPENDS ON N )
                                                                  #/
       N>=2;
     /# TERMINATION OF LOG(2,N)
       SOME D FIXED.(D>=0 & N<=D) BY INTRO,N;
       N>=0; N>=N;
       LOG(2,N)>=1 BY ALLEL,LOG_POSITIVE,2,N;
```



```
1146      0<U-L BY ARITH,U-L>=LOG(2,N)>=1;
1147      L<U BY ARITH,0<U-L,+,L=L;
1148      L+1<=HBOUND(A,1) BY ARITH,L+1<=U,DOM(A,U);
1149      /# L<L+1 BY ARITH;  DO WE NEED THIS                        #/
1149      LBOUND(A,1)<=L+1 BY ARITH,LBOUND(A,1)<=L,L<L+1;
1150      DOM(A,L+1);                                               #/
1151  /# SHOW N/P IS GREATER THAN 1
1151      P<=N BY ALLEL,DIVISOR_SIZE,P,N;
1152      N/P >= 1 BY ALLEL,DIVISION_LEMMA,N,P; /# NEED THE LEMMA    #/
1153      N/P > 1 BY ARITH,N/P>=1,N/P^=1;                           #/
1154  /# SHOW U-(L+1)<D-1 FOR TERMINATION
1154      U-(L+1) < D-1 BY ARITH,U-L<D,-,1=1;
1155  */
1155       /* USE AR TO NAME A BEFORE ASSIGNMENT A(L)=P              *
1155  /*: AR = A;                                                   *
          /*/ AR = A; AR = AA;
1156  N2 = N/P;  A(L) = P;
1158  /*/ N2 = N/P; N2>1; U-(L+1)>=LOG(2,N2);  A(L) = P;            *
1160       /* BY ARRAY ASSIGNMENT:                                 *
1164       /*/ ASGN2: ALL I FIXED WHERE I^=L.AR(I)=A(I);          *
1164       /* USE AAA TO KEEP TRACK OF THE AFFECT                  *
1165       /* OF PRIME_FACTORIZATION ON A.                         *
1165       /*: AAA = A;                                           *
1165       /*/ TRANSF: ALL I FIXED   . AAA(I) = A(I) BY INTRO;
1166       AAA = A;
1167       AAA(L) = A(L) BY ALLEL,TRANSF,L;
1168       AAA(L) = P;                                             #
1169            /# RELATE AA TO AAA FOR PROOF OF AT4 BELOW
1170            AA_AAA: ALL I FIXED WHERE I^=L.AA(I) = AAA(I)
1170            BY INTRO,PROOF;
1170              AR(I) = A(I) BY ALLEL,ASGN2,I;
1171              AA(I) = AR(I);
1172            QED;
1173       LBOUND(A,1) = LBOUND(AAA,1);
1174       HBOUND(A,1) = HBOUND(AAA,1);
1175       DOM(AAA,L+1); DOM(AAA,U);
1176   CALL PRIME_FACTORIZATION( (N2) ,A,L+1,(U),M /*:,(AAA) */);
1178       /* LIST CONSEQUENCES
1179       /*/
1179       C1: N2 = PROD(A,L+1,M);
1179       C2: L+1<=M<=U;  L<=M; DOM(A,M);
1180       C3: ALL I FIXED WHERE L+1<=I<=M.(PRIME(A(I)) & DIV(A(I),N2));
1183       C4: ALL (I,J) FIXED WHERE L+1<=I<=J<=M. A(I)<=A(J);
1184       C5:  /# TRANSFER CONDITION ALLOWING PROOF THAT A(L) IS UNCHANGED
1185            ALL I FIXED WHERE LBOUND(A,1)<=I<L+1.AAA(I)=A(I);
1185            LBOUND(A,1)<=L<L+1 ;
1186            AAA(L) = A(L) BY ALLEL,C5,L;
1187       A(L) = AAA(L) = P;
1188       N = P*(N/P) + MOD(N,P) BY FUNCTION,DIVISION(N,P);
1189       N = A(L)*N2 + 0;  N = A(L)*N2;
1190       N = A(L)*PROD(A,L+1,M);
1192       SOME D FIXED.(D>=0 & MONUS(M+1,L)<D) BY FUNCTION,
1193       PROD_TERMINATION(A,L,M);
1193
```

```
     PROD(A,L,M) = A(L)*PROD(A,L+1,M) BY FUNCTION,PROD(A,L,M);
     */
/*/ /# ******* PROVE OUTPUT PROPERTIES *********                    #/
/# ALL OUTPUT PROPERTIES EXCEPT THE ORDER OF THE F ARE               #/
/# EASY TO PROVE, FOR EXAMPLE,                                      #/
     SOME D FIXED.(D>=0 & MONUS(U+1,L)<D) BY FUNCTION,
     PROD_TERMINATION(A,L,U);
PROD(A,L,U) = A(L)*PROD(A,L+1,U) BY FUNCTION, PROD(A,L,U);
M <= U;
/# WE NOW PROVE THAT THE FACTORS ARE ORDERED.                        #/
ALL (I,J) FIXED WHERE L<=I<=J<=M. A(I) <= A(J) BY INTRO,
PROOF;
/# FOR ANY J IN L<=J<=M A(J) IS A PRIME FACTOR OF N, BUT             #/
/# A(L) IS THE LEAST PRIME FACTOR, SO THE RESULT F FROM C4.         #/
D1: L+1<=I | I=L BY ARITH, L<=I;
A(I) <= A(J) BY CASES,D1,
PROOF;
  CASE L+1<=I;
  L+1<=I BY ARITH,L+1<=I,L<=I<=J<=M;
     I<=M BY ARITH,L<=I<=J<=M;
  A(I) <= A(J) BY ALLEL,C4,I,J;
  CASE I=L;
  /# USE THE FACT THAT A(L) IS THE LEAST PRIME FACTOR.              #/
  D2: A(I) > A(J) | A(I) <= A(J) BY ARITH;
  A(I) <= A(J) BY CASES,D2,
  PROOF;
  CASE A(J) < A(I);
     ( I=J | I^=J ) BY ARITH;
    I^=J BY CASES,(I=J | I^=J),
    PROOF;CASE I=J;A(I)=A(J);'0'B BY ARITH,A(I)=A(J),A(J)<A(I); QED;
    L+1<=J BY ARITH,L<=I<=J,I^=J;
    PRIME(A(J)) & DIV(A(J),N2) BY ALLEL,C3,J;
    N = P*N2;  N = N2*P;
    DIV(A(J),N) BY ALLEL,DIVIDE_PRODUCT,A(J),N2,P;
    1<A(J); 1<A(J)<A(I);  A(I) = A(L) = P; 1<A(J)<P;
    ^DIV(A(J),N) BY ALLEL,ALL I FIXED WHERE 1<I<P.^DIV(I,N),A(J);
    '0'B; /# A(J) BOTH DIVIDES AND DOES NOT DIVIDE N               #/
    QED; /# END OF CASES ON D2                                     #/
QED; /# END OF CASES ON D1                                         #/
QED; /# END OF INTRO                                               #/
/# NOW ROUTINELY EXTEND THE DOMAIN OF C3.                          #/
ALL I FIXED WHERE L<=I<=M.( (PRIME(A(I)) & DIV(A(I),N)) ) BY INTRO,
PROOF;
  D1: L+1<=I | L=I BY ARITH, L<=I;
  PRIME(A(I)) & DIV(A(I),N) BY CASES,D1,
  PROOF;
     CASE L+1<=I;
     L+1<=I<=M;
     PRIME(A(I)) & DIV(A(I),N2) BY ALLEL,C3,I;
     N = P*N2;  N = N2*P;
     DIV(A(I),N) BY ALLEL,DIVIDE_PRODUCT,A(I),N2,P;
     CASE L=I; DIV(P,N); DIV(A(L),N); DIV(A(I),N); PRIME(A(I));
  QED; /# END OF CASES ON D1                                       #/
```



```
1244      QED; /# END OF BOUNDED ALL INTRO                                 #/
1245           /# FINALLY PROVE THE TRANSFER CONDITION, AT4                 #/
1245           /# BY RELATING A BEFORE A(L)=P AND CALL OF                   #/
1245           /# PRIME_FACTORIZATION TO A AFTER.                           #/
1245           ALL I FIXED WHERE LBOUND(A,1)<=I<L.AA(I)=A(I)
1245           BY INTRO,PROOF;
1246             I<L+1 BY ARITH,I<L; I^=L BY ARITH,I<L;
1248             AAA(I) = A(I) BY ALLEL,C5,I;
1249             AAA(I) = AA(I) BY ALLEL,AA_AAA,I;
1250
1250           QED;                                                         */
1251      RETURN;
1252    END PRIME_FACTORIZATION;
*THEOREM
1253      /*/ FOR (N,F(*),L,U) FIXED DEFINE PRIME_FACTORS(N,F,L,U) =
1253           ALL I FIXED WHERE L<=I<=U.(PRIME(F(I)) & DIV(F(I),N) );
1254          FOR (F(*),L,U) FIXED DEFINE ORDERED(F,L,U) =
1254           ALL (I,J) FIXED WHERE L<=I<=J<=U. F(I)<=F(J);
1255          FOR (N,F(*),L,U) FIXED DEFINE FACTORIZATION(N,F,L,U) =
1255           N = PROD(F,L,U) & L<=U & PRIME_FACTORS(N,F,L,U) & ORDERED(F,L,U);
1256      */
*PROCESS
1256      FTA_LEMMA: PROCEDURE(N,P,F,L,U) RETURNS(BIT(1));
1257           DECLARE (N,P,F(*),L,U) FIXED  ;
1257           /* A LEMMA NEEDED TO SHOW THAT PRIME FACTORIZATION IS UNIQUE    */
1258           /*/ ASSUME N > 1, PRIME(P), DIV(P,N),
1258                DOMAIN: DOM(F,L) & DOM(F,U),
1258                FACT: FACTORIZATION(N,F,L,U),
1258                LESS: ALL I FIXED WHERE L<=I<=U. P < F(I);              */
1259           /*/ DEFINE FALSE = '0'B;                                      */
1260           /*/ ATTAIN FALSE;                                             */
1261           /* SHOW THAT SINCE P DIVIDES N AND IS PRIME, IT MUST DIVIDE    */
1261           /* ONE OF THE F(I) ( A CONSEQUENCE OF PRIME_DIVIDES_LONG_PRODUCT*/
1261           /* SINCE F(I) IS ALSO PRIME, P MUST EQUAL F(I).  BUT THIS IS   */
1261           /* IMPOSSIBLE SINCE P<F(I).                                    */
1261           /*/  /# ***** BODY OF THE PROOF *****                         #/
1261           L<=U;
1262           DIV(P,PROD(F,L,U)); /# FROM FACTORIZATION AND DIV(P,N)        #/
1263           ALL I FIXED WHERE L<=I<=U.F(I)>0 BY INTRO,
1263           PROOF;
1264             L<=I<=U; /# DEFINITION OF DOM(F,I) AND DOMAIN ASSUME        #/
1265             PRIME(F(I)) BY ALLEL,PRIME_FACTORS(N,F,L,U),I;
1266             F(I)>0 BY ARITH,F(I)>1; /# FROM DEF OF PRIME                #/
1267           QED;
1268           DIV(P,PROD(F,L,U));
1269           STMT2(U-L)
1269           BY ALLEL,PRIME_DIVIDES_LONG_PRODUCT,U-L;
1270           /# POOR STYLE IN WRITING THE QUANTIFIERS IN                   #/
1270           /# PRIME_DIVIDES_LONG_PRODUCT RESULTS IN ALL                 #/
1270           /# THIS FUSS TO GET AT THE STATEMENT.                        #/
1270           /# WE NEED TO BE CAREFUL ABOUT CAPTURING U-L                 #/
1270           /# SO WE RENAME THE BOUND VARIBLES OF STMT2(U-L)             #/
1270           LA: ALL (A(*),LL,UU) FIXED WHERE LL<=UU & DOM(A,LL) &
```