Acta Informatica 22, 699–707 (1986)

Erratum

Proving Properties of Pascal Programs in MIZAR 2

Piotr Rudnicki and Włodzimierz Drabent

Acta Informatica 22, 311–331 (1985)

Due to an unfortunate error, Figs. 1–6 and Appendices 1 and 2 were omitted:

5. A bit of Dessert

The list reversing program is very often used by programmers, which included the authors. It is usually understood that the source list is a *nil* ending list; the program reverses it as proven. The work on proving the program has given us a new motive to prove programs at all. This is: while proving a program property we have a chance to observe what the program can actually do.

While proving the list reversing program we have realized that it processes also cyclic lists, terminates and results in changing their orientation; but that is not proven yet. So we have:

1. For a linear list

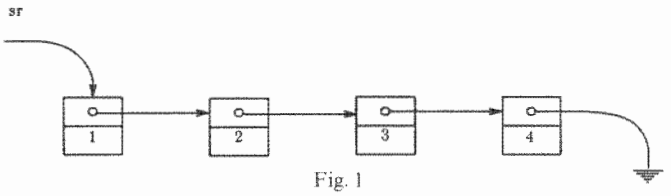

Fig. 1

we obtain

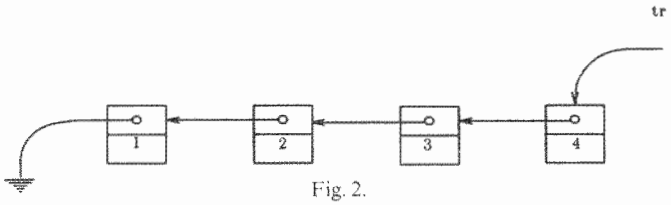

Fig. 2.

2. For a cyclic list

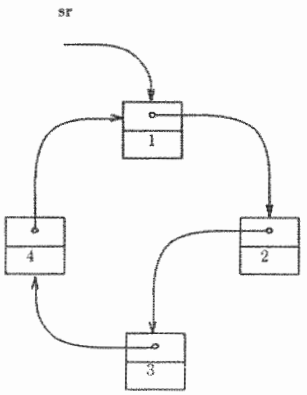

Fig. 3 we obtain

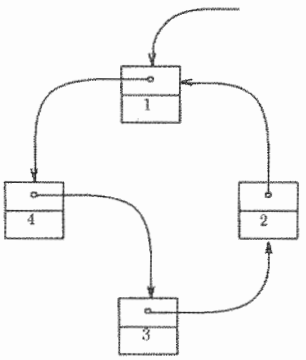

Fig. 4.

3. For a cyclic list with a "stem"

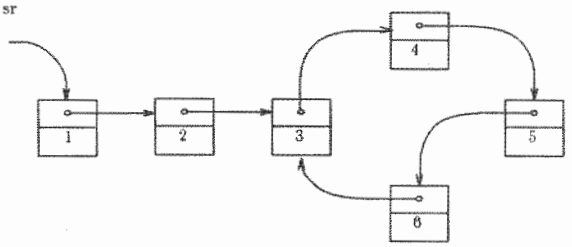

Fig. 5.

the program also terminates, and in this case we obtain

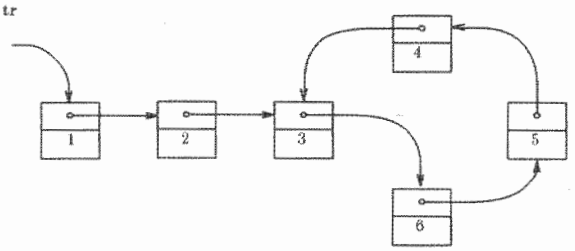

Fig. 6.

Of the programmers known to us and using that program none was conscious of this fact.

Comment on Appendices

The first (marked with dashes) part of a MIZAR 2 text is called an environment and contains definitions and axioms. The lines following the heading 'Program description' are presumed to be generated by a describer. The lines following the heading 'Description technology' are prepared once and by hand, and later on are appended (possibly in parts) to a particular problem. The fragments of environments pertinent to a particular domain, in our case to arithmetic and list theory, are to be prepared by hand (and may be used in many proofs). The actual proof follows the symbol *BEGIN* and is done by hand. The part of a line starting with == constitutes an informal commentary. The final *THANKS O.K.* is the checker signal confirming correctness of the proof.

Appendix 1

The proof for the factorial computing program in MIZAR 2

```
ENVIRON

== D E S C R I P T I O N    T E C H N O L O G Y ============

TYPE HISTORY;
LET H,H' DENOTE HISTORY;
GIVEN FH BEING HISTORY;
FOR H CONSIDER NH BEING HISTORY;
CONSIDER FINISH BEING NATURAL;
FOR H CONSIDER CP BEING NATURAL;
FOR M BEING NATURAL RECONSIDER +M AS ELEMENT OF INTEGERS;
LET K, L, N DENOTE ELEMENT OF INTEGERS;
FOR K, L RECONSIDER K+L, K*L AS ELEMENT OF INTEGERS;
FOR K,L PRED K (= L;
FOR A BEING NONEMPTY CONSIDER PTR BEING NONEMPTY;
DEFINITION LET A BE NONEMPTY, V BE (ELEMENT OF PTR(A)),
                H BE HISTORY, C BE ELEMENT OF A;
  PRED C = VAL(V,H)  END;

== A R I T H M E T I C S ========================================

SCHEME INDUCTION;  PRED P;
 FOR N  ST +0 (= N HOLDS P[N]
  SINCE
   COND1: P[+0];
   COND2: FOR N  ST +0 (= N & P[N] HOLDS P[N+(+1)]    END;
FOR K CONSIDER FACTORIAL BEING ELEMENT OF INTEGERS;
AR1: FACTORIAL(+0) = +1 ;
AR2: FOR K, N ST N+(+1) (= K HOLDS (N (= K & N () K);
AR3: FOR N HOLDS  N(=N ;
AR4: FOR N ST +0 (= N
        HOLDS FACTORIAL(N)*(N+(+1))=FACTORIAL(N+(+1));

== P R O G R A M    D E S C R I P T I O N ==================

LET V DENOTE ELEMENT OF PTR(INTEGERS);
                    == VAR X, Y, Z: INTEGER;
CONSIDER X,Y,Z BEING ELEMENT OF PTR(INTEGERS);
                    ==  BEGIN (* START: *) Y := 1;
AX01: CP(NH(FH)) = 1;
AX02: VAL(Y,NH(FH)) = +1;
AX03: VAL(Z,NH(FH)) = VAL(Z,FH);
                    ==    (* 1: *) Z := 0;
AX11: CP(NH(NH(FH))) = 2;
AX12: VAL(Z,NH(NH(FH))) = +0;
AX13: VAL(Y,NH(NH(FH))) = +1;
                    ==    (* 2: *) WHILE Z () X DO BEGIN
AX21: FOR H ST CP(H) = 2 HOLDS
          (VAL(Z,H) () VAL(X,H) IMPLIES CP(NH(H)) = 3) &
          (VAL(Z,H)  = VAL(X,H) IMPLIES CP(NH(H)) = FINISH);
AX22: FOR H,V ST CP(H) = 2 HOLDS VAL(V,NH(H)) = VAL(V,H);
                    ==    (* 3: *)     Z := Z+1;
AX31: FOR H ST CP(H)=3 HOLDS CP(NH(H))=4;
AX32: FOR H ST CP(H)=3 HOLDS VAL(Z,NH(H))=VAL(Z,H)+(+1);
AX33: FOR H,V ST CP(H)=3 & V()Z HOLDS VAL(V,NH(H))=VAL(V,H);
                    ==    (* 4: *)     Y := Y*Z      END
AX41: FOR H ST CP(H)=4 HOLDS CP(NH(H))=2;
AX42: FOR H ST CP(H)=4 HOLDS VAL(Y,NH(H))=VAL(Y,H)*VAL(Z,H);
```

```
-   AX43: FOR H,V ST CP(H)=4 & V()Y HOLDS VAL(V,NH(H))=VAL(V,H);
-                    ==    (* FINISH: *) END.
-   AYZ: Y()Z;
-   AXVAL: FOR H HOLDS VAL(X,H) = VAL(X,FH);
-
-   == P R O P E R T I E S ==========================================
-
-   BEGIN
-
-    FIRSTCONDITION:
     NOW
      ASSUME  +0 (= VAL(X,FH);
       VAL(Y,NH(NH(FH))) = FACTORIAL(+0) BY AR1,AX13;
     HENCE EX H ST CP(H) = 2 & VAL(Z,H) = +0  &
                   VAL(Y,H) = FACTORIAL(+0) BY AX11,AX12
     END;

    SECONDCONDITION:
     NOW LET N BE ELEMENT OF INTEGERS SUCH THAT
      A0: +0 (= N AND
      A1: N (= VAL(X,FH) IMPLIES
                (EX H ST CP(H) = 2 &
                   VAL(Z,H) = N & VAL(Y,H) = FACTORIAL(N));
     ASSUME A2: N+(+1) (= VAL(X,FH);
     THUS EX H ST CP(H)=2 & VAL(Z,H) = N+(+1) &
                   VAL(Y,H) = FACTORIAL(N+(+1))
      PROOF
       N (= VAL(X,FH) BY AR2,A2;
       THEN CONSIDER H SUCH THAT
       A3: CP(H)=2 & VAL(Z,H)=N & VAL(Y,H)=FACTORIAL(N) BY A1;
       N () VAL(X,FH) BY AR2,A2;
       THEN VAL(Z,H) () VAL(X,H) BY AXVAL,A3;
       THEN CP1: CP(NH(H)) = 3 BY AX21,A3;
       THEN CP2: CP(NH(NH(H))) = 4 BY AX31;
       THEN CP3: CP(NH(NH(NH(H)))) = 2 BY AX41;
       Z1: VAL(Z,NH(H)) = VAL(Z,H) BY AX22,A3 .= N BY A3;
       Z2: VAL(Z,NH(NH(H))) = VAL(Z,NH(H))+(+1) BY AX32,CP1
                            .= N+(+1) BY Z1;
       Z3: VAL(Z,NH(NH(NH(H)))) = VAL(Z,NH(NH(H)))
                                                BY AX43,CP2,AYZ
                                .= N+(+1) BY Z2;
       Y1: VAL(Y,NH(NH(H))) = VAL(Y,NH(H)) BY AX33,CP1,AYZ
                             .= VAL(Y,H) BY AX22,A3
                             .= FACTORIAL(N) BY A3;
           VAL(Y,NH(NH(NH(H))))=VAL(Y,NH(NH(H)))*VAL(Z,NH(NH(H)))
                                                    BY AX42,CP2
                            .= FACTORIAL(N)*(N+(+1)) BY Z2,Y1
                            .= FACTORIAL(N+(+1)) BY AR4,A0;
           HENCE EX H' ST CP(H') = 2 & VAL(Z,H') = N+(+1) &
                          VAL(Y,H') = FACTORIAL(N+(+1)) BY CP3,Z3
         END
      END; == SECONDSTEP

      LEMMA:
       FOR N ST +0 (= N HOLDS N (= VAL(X,FH) IMPLIES
         (EX H ST CP(H)=2 & VAL(Z,H)=N & VAL(Y,H)=FACTORIAL(N))
               FROM INDUCTION(FIRSTCONDITION,SECONDCONDITION);

      FACTPROGPROPERTY:
       +0 (= VAL(X,FH) IMPLIES
         (EX H ST CP(H)=FINISH & VAL(Y,H)=FACTORIAL(VAL(X,FH)))
        PROOF
         ASSUME A1: +0 (= VAL(X,FH);
          VAL(X,FH) (= VAL(X,FH) BY AR3;
```

```
          THEN CONSIDER H SUCH THAT
                A2: CP(H) = 2 AND
                A3: VAL(Z,H) = VAL(X,FH) AND
                A4: VAL(Y,H) = FACTORIAL(VAL(X,FH)) BY A1,LEMMA;
         TAKE H' = NH(H);
          Z1: VAL(Z,H) = VAL(X,FH) BY A3 .= VAL(X,H) BY AXVAL;
         HENCE CP(H') = CP(NH(H)) .= FINISH BY AX21,A2,Z1;
          VAL(Y,NH(H)) = VAL(Y,H) BY AX22,A2;
         HENCE VAL(Y,H') = FACTORIAL(VAL(X,FH)) BY A4
        END;
    END

THANKS,O.K
----------
```

Appendix 2

The proof for the list reversing program in MIZAR 2

```
ENVIRON

== D E S C R I P T I O N   T E C H N O L O G Y ==============
TYPE HISTORY;  LET  H,H',H'' DENOTE HISTORY;
GIVEN FH BEING HISTORY;
FOR H CONSIDER NH BEING HISTORY;
FOR H CONSIDER CP BEING NATURAL;
CONSIDER FINISH BEING NATURAL;
LET A DENOTE NONEMPTY;
FOR A CONSIDER PTR BEING NONEMPTY;
FOR A CONSIDER NIL BEING ELEMENT OF PTR(A);
DEFINITION LET A BE NONEMPTY, V BE(ELEMENT OF PTR(A)),
              H BE HISTORY, C BE ELEMENT OF A;
  PRED C=VAL(V,H)  END;
DEFINITION LET A,B BE NONEMPTY, V BE (ELEMENT OF PTR(A)),
              Y BE (ELEMENT OF PTR(B)), N BE NATURAL;
  PRED Y=FIELD(A,V,N,B)  END;

== L I S T   T H E O R Y ====================================
TYPE LIST OF A BEING NONEMPTY;
FOR A CONSIDER NILL BEING LIST OF A;
FOR A BEING NONEMPTY, E BEING (ELEMENT OF A),
    L BEING LIST OF A
   RECONSIDER E.L AS LIST OF A;
LIST1: FOR A BEING NONEMPTY, E BEING (ELEMENT OF A),
          L BEING LIST OF A
         HOLDS  E.L()NILL(A);
SCHEME LISTIND; CONST A BEING NONEMPTY; PRED P;
  FOR L BEING LIST OF A HOLDS P[L]
   SINCE
    COND1: P[NILL(A)];
    COND2: FOR E BEING (ELEMENT OF A), L BEING LIST OF A
             ST P[L] HOLDS P[E.L]    END;
FOR A BEING NONEMPTY, L,L' BEING LIST OF A
   RECONSIDER L+L' AS LIST OF A;
CONC1: FOR A BEING NONEMPTY, L BEING LIST OF A
         HOLDS NILL(A)+L=L & L+NILL(A)=L;
DEFINITION LET A BE NONEMPTY, L,L' BE LIST OF A;
  PRED L'=REV(L)  END;
REV1: FOR A BEING NONEMPTY HOLDS REV(NILL(A))=NILL(A);
```

```
-   REVLEMMA:
-      FOR A BEING NONEMPTY, E BEING (ELEMENT OF A),
-          L,L' BEING LIST OF A
-        HOLDS REV(E.L)+L'=REV(L)+(E.L');
-   DEFINITION LET A BE NONEMPTY, E BE (ELEMENT OF A),
-                  L BE LIST OF A;
-     PRED E IS MEMBER OF L  END;
-   FOR A BEING NONEMPTY, L,L' BEING LIST OF A
-      PRED L MISSES L' DENOTES NOT (EX X BEING ELEMENT OF A ST
-                        X IS MEMBER OF L & X IS MEMBER OF L');
-   AX1LT: FOR A BEING NONEMPTY, L,L' BEING LIST OF A
-           ST NILL(A)=L HOLDS L' MISSES L ;
-
-   == P R O G R A M    D E S C R I P T I O N ==================
-     ====TYPE LREC=RECORD NEXT: ^LREC; CONT: INTEGER END;
-   GIVEN LREC BEING NONEMPTY;
-   FOR V BEING ELEMENT OF PTR(LREC)
-     TAKE NEXT=FIELD(LREC,V,1,PTR(LREC)),
-          CONT=FIELD(LREC,V,2,INTEGERS);
-
-   == L I S T    R E P R E S E N T A T I O N ==================
-   DEFINITION LET V BE (ELEMENT OF PTR(PTR(LREC))),
-                  L BE (LIST OF PTR(LREC)), H BE HISTORY;
-     PRED L=LIST(V,H)  END;
-   LET V DENOTE ELEMENT OF PTR(PTR(LREC));
-   LET E DENOTE ELEMENT OF PTR(LREC);
-   LET L, L' DENOTE LIST OF PTR(LREC);
-   REPS1: FOR H,V HOLDS
-            NILL(PTR(LREC))=LIST(V,H) IFF VAL(V,H)=NIL(LREC);
-   REPS2: FOR E, L, H, V HOLDS E.L=LIST(V,H) IFF
-                      (E=VAL(V,H) & L=LIST(NEXT(VAL(V,H)),H));
-   REPS3: FOR H,V ST VAL(V,H)<>NIL(LREC)
-            HOLDS VAL(V,H) IS MEMBER OF LIST(V,H);
-   AX1LR: FOR H,V HOLDS
-            NOT VAL(V,H) IS MEMBER OF LIST(NEXT(VAL(V,H)),H);
-   TH1LR: FOR L, L', E ST E.L MISSES L' HOLDS L MISSES E.L';
-
-   == P R O G R A M   D E S C R I P T I O N   CONTINUED =======
-     ====VAR LAUX,SR,TR: ^LREC;
-   CONSIDER LAUX,SR,TR BEING ELEMENT OF PTR(PTR(LREC));
-   AX1: LAUX<>SR; AX2: LAUX<>TR; AX3: SR<>TR;
-   AX4: FOR H, V HOLDS SR<>NEXT(VAL(V,H)) &
-                    TR<>NEXT(VAL(V,H)) & LAUX<>NEXT(VAL(V,H));
-     ====BEGIN (*CP  START*) TR :=NIL;
-   AX01: CP(NH(FH))=1;
-   AX02: VAL(TR,NH(FH))=NIL(LREC);
-   AX03: FOR V ST V<>TR HOLDS VAL(V,NH(FH))=VAL(V,FH);
-   AX04: FOR V ST V<>TR HOLDS LIST(V,NH(FH))=LIST(V,FH);
-     ====(*CP 1*) WHILE SR<>NIL DO BEGIN
-   AX11: FOR H ST CP(H)=1 HOLDS
-           (VAL(SR,H)<>NIL(LREC) IMPLIES CP(NH(H))=2) &
-           (VAL(SR,H) =NIL(LREC) IMPLIES CP(NH(H))=FINISH);
-   AX12: FOR A BEING NONEMPTY, V BEING (ELEMENT OF PTR(A)), H
-            ST CP(H)=1 HOLDS VAL(V,NH(H))=VAL(V,H);
-   AX13: FOR H,V ST CP(H)=1 HOLDS LIST(V,NH(H))=LIST(V,H);
-     ====(*CP 2*)  LAUX :=SR^.NEXT;
-   AX21: FOR H ST CP(H)=2 HOLDS CP(NH(H))=3;
-   AX22: FOR H ST CP(H)=2
-            HOLDS VAL(LAUX,NH(H))=VAL(NEXT(VAL(SR,H)),H);
-   AX23: FOR A BEING NONEMPTY, V BEING (ELEMENT OF PTR(A)), H
-            ST CP(H)=2 & V<>LAUX HOLDS VAL(V,NH(H))=VAL(V,H);
-   AX24: FOR V,H ST CP(H)=2 & V<>LAUX
-            HOLDS LIST(V,NH(H))=LIST(V,H);
```

```
-    AX25: FOR H ST CP(H)=2
-            HOLDS LIST(LAUX,NH(H))=LIST(NEXT(VAL(SR,H)),H);
-      ====(*CP  3*)  SR^.NEXT :=TR;
-    AX31: FOR H ST CP(H)=3 HOLDS CP(NH(H))=4;
-    AX32: FOR H ST CP(H)=3
-            HOLDS VAL(NEXT(VAL(SR,H)),NH(H))=VAL(TR,H);
-    AX33: FOR A BEING NONEMPTY, V BEING (ELEMENT OF PTR(A)), H
-            ST CP(H)=3 & V()NEXT(VAL(SR,H))
-              HOLDS VAL(V,NH(H))=VAL(V,H);
-    AX34: FOR H
-           ST CP(H)=3 & NOT VAL(SR,H) IS MEMBER OF LIST(TR,H)
-            HOLDS LIST(NEXT(VAL(SR,H)),NH(H))=LIST(TR,H);
-    AX35: FOR H,V
-            ST CP(H)=3 & NOT VAL(SR,H) IS MEMBER OF LIST(V,H)
-             HOLDS LIST(V,NH(H))=LIST(V,H);
-      ====(*CP  4*)  TR :=SR;
-    AX41: FOR H ST CP(H)=4 HOLDS CP(NH(H))=5;
-    AX42: FOR H ST CP(H)=4 HOLDS VAL(TR,NH(H))=VAL(SR,H);
-    AX43: FOR A BEING NONEMPTY, V BEING (ELEMENT OF PTR(A)), H
-            ST CP(H)=4 & V()TR HOLDS VAL(V,NH(H))=VAL(V,H);
-    AX44: FOR H ST CP(H)=4 HOLDS LIST(TR,NH(H))=LIST(SR,H);
-    AX45: FOR V,H ST CP(H)=4 & V()TR
-            HOLDS LIST(V,NH(H))=LIST(V,H);
-      ====(*CP 5*) SR :=LAUX
-    AX51: FOR H ST CP(H)=5 HOLDS CP(NH(H))=1;
-    AX52: FOR H ST CP(H)=5 HOLDS VAL(SR,NH(H))=VAL(LAUX,H);
-    AX53: FOR A BEING NONEMPTY, V BEING (ELEMENT OF PTR(A)), H
-            ST CP(H)=5 & V()SR HOLDS VAL(V,NH(H))=VAL(V,H);
-    AX54: FOR H ST CP(H)=5 HOLDS LIST(SR,NH(H))=LIST(LAUX,H);
-    AX55: FOR V,H ST CP(H)=5 & V()SR
-            HOLDS LIST(V,NH(H))=LIST(V,H);
-      ====                              END (*CP  FINISH*)
-    BEGIN
-     FIRSTCOND:
      NOW LET H BE HISTORY SUCH THAT
       A1: CP(H)=1 & NILL(PTR(LREC))=LIST(SR,H) &
                              NILL(PTR(LREC)) MISSES LIST(TR,H);
         S1: VAL(SR,H)=NIL(LREC) BY A1,REPS1;
         LIST(TR,H)=NILL(PTR(LREC))+LIST(TR,H) BY CONC1
                    .=REV(NILL(PTR(LREC)))+LIST(TR,H) BY REV1;
       HENCE EX H' ST CP(H')=1 & VAL(SR,H')=NIL(LREC) &
                      LIST(TR,H')=REV(NILL(PTR(LREC)))+LIST(TR,H)
                                                        BY A1,S1
      END; ==FIRSTCOND
     SECONDCOND:
      NOW LET E BE (ELEMENT OF PTR(LREC)),
              L BE LIST OF PTR(LREC) SUCH THAT
      A1: FOR H BEING HISTORY
             ST CP(H)=1 & L=LIST(SR,H) & L MISSES LIST(TR,H)
               EX H' ST CP(H')=1 & VAL(SR,H')=NIL(LREC) &
                                  LIST(TR,H')=REV(L)+LIST(TR,H);
      LET H BE HISTORY SUCH THAT
       A2: CP(H)=1 & E.L=LIST(SR,H) & E.L MISSES LIST(TR,H);
      THUS EX H'' ST CP(H'')=1 & VAL(SR,H'')=NIL(LREC) &
                               LIST(TR,H'')=REV(E.L)+LIST(TR,H)
       PROOF
        E.L()NILL(PTR(LREC)) BY LIST1;
        THEN LIST(SR,H)()NILL(PTR(LREC)) BY A2;
        THEN S1: VAL(SR,H)()NIL(LREC) BY REPS1;
        THEN CP2: 2=CP(NH(H)) BY AX11,A2;
        THEN CP3: 3=CP(NH(NH(H))) BY AX21;
                  TAKE H4=NH(NH(NH(H)));
             CP4: 4=CP(NH(NH(NH(H)))) BY AX31,CP3 .=CP(H4);
        THEN CP5: 5=CP(NH(H4)) BY AX41;
        THEN CP6: 1=CP(NH(NH(H4))) BY AX51;
```

```
    S2: VAL(SR,NH(H))=VAL(SR,H) BY AX12,A2;
    S3: VAL(SR,NH(NH(H)))=VAL(SR,NH(H)) BY AX23,AX1,CP2
                        .=VAL(SR,H) BY S2;
        SR()NEXT(VAL(SR,NH(NH(H)))) BY AX4; THEN
    S4: VAL(SR,NH(NH(H)))=VAL(SR,NH(NH(NH(H)))) BY AX33,CP3
                        .=VAL(SR,H4);
    S5: LIST(SR,H)=LIST(SR,NH(H)) BY AX13,A2
                  .=LIST(SR,NH(NH(H))) BY AX24,CP2,AX1;
    T1: LIST(TR,H)=LIST(TR,NH(H)) BY AX13,A2
                  .=LIST(TR,NH(NH(H))) BY AX24,CP2,AX2;
    L1: LIST(LAUX,NH(NH(H)))
             =LIST(NEXT(VAL(SR,NH(H))),NH(H)) BY AX25,CP2
            .=LIST(NEXT(VAL(SR,H)),NH(H)) BY S2
            .=LIST(NEXT(VAL(SR,H)),H) BY AX13,A2;
        NOT VAL(SR,H) IS MEMBER OF LIST(NEXT(VAL(SR,H)),H)
                                              BY AX1LR; THEN
    M1: NOT VAL(SR,NH(NH(H)))
             IS MEMBER OF LIST(LAUX,NH(NH(H))) BY L1,S3;
    S6: LIST(SR,NH(NH(H4)))=LIST(LAUX,NH(H4)) BY AX54,CP5
                       .=LIST(LAUX,H4) BY AX45,AX2,CP4
                       .=LIST(LAUX,NH(NH(NH(H))))
                       .=LIST(LAUX,NH(NH(H))) BY AX35,CP3,M1
                       .=LIST(NEXT(VAL(SR,H)),H) BY L1
                       .=L BY REPS2,A2;
        LIST(SR,NH(NH(H)))
            MISSES LIST(TR,NH(NH(H))) BY S5,T1,A2; THEN
    M2: NOT (EX X BEING ELEMENT OF PTR(LREC) ST
                        X IS MEMBER OF LIST(SR,NH(NH(H))) &
                        X IS MEMBER OF LIST(TR,NH(NH(H))));
        VAL(SR,NH(NH(H)))()NIL(LREC) BY S1,S3; THEN
        VAL(SR,NH(NH(H)))
            IS MEMBER OF LIST(SR,NH(NH(H))) BY REPS3; THEN
    M3: NOT VAL(SR,NH(NH(H)))
                     IS MEMBER OF LIST(TR,NH(NH(H))) BY M2;
    S7: LIST(NEXT(VAL(SR,H4)),H4)
        =LIST(NEXT(VAL(SR,NH(NH(H)))),NH(NH(NH(H)))) BY S4
       .=LIST(TR,NH(NH(H))) BY AX34,M3,CP3
       .=LIST(TR,H) BY T1;
    S8: E=VAL(SR,H) BY A2,REPS2 .= VAL(SR,H4) BY S4,S3;
    T2: LIST(TR,NH(NH(H4)))
                  =LIST(TR,NH(H4)) BY AX55,AX3,CP5
                 .=LIST(SR,H4) BY AX44,CP4
                 .=E.LIST(TR,H) BY REPS2,S7,S8;
        L MISSES E.LIST(TR,H) BY TH1LR,A2;
        THEN L MISSES LIST(TR,NH(NH(H4))) BY T2; THEN
    CONSIDER H'' BEING HISTORY SUCH THAT
    A3: CP(H'')=1 & VAL(SR,H'')=NIL(LREC) &
               LIST(TR,H'')=REV(L)+LIST(TR,NH(NH(H4)))
                                                 BY A1,CP6,S6;
    REV(L)+LIST(TR,NH(NH(H4)))=REV(L)+(E.LIST(TR,H)) BY T2
                          .=REV(E.L)+ LIST(TR,H) BY REVLEMMA;
    HENCE CP(H'')=1 & VAL(SR,H'')=NIL(LREC) &
               LIST(TR,H'')=REV(E.L)+LIST(TR,H) BY A3
  END
 END;==SECONDCOND

LOOP:
 FOR L BEING(LIST OF PTR(LREC)), H
   ST CP(H)=1 & L=LIST(SR,H) & L MISSES LIST(TR,H)
     EX H' ST CP(H')=1 & VAL(SR,H')=NIL(LREC) &
               LIST(TR,H')=REV(L)+LIST(TR,H)
                     FROM LISTIND(FIRSTCOND,SECONDCOND);
```

```
    T1: NILL(PTR(LREC))=LIST(TR,NH(FH)) BY AX02,REPS1;
    S1: LIST(SR,NH(FH)) MISSES LIST(TR,NH(FH)) BY T1,AX1LT;
        LIST(SR,NH(FH))=LIST(SR,NH(FH));  ==ACHILLES' HEEL
    THEN CONSIDER H' BEING HISTORY SUCH THAT
     A1: CP(H')=1 & VAL(SR,H')=NIL(LREC) &
              LIST(TR,H')=REV(LIST(SR,NH(FH)))+LIST(TR,NH(FH))
                                                BY LOOP,AX01,S1;
    C1: CP(NH(H'))=FINISH BY AX11,A1;
    C2: REV(LIST(SR,FH))=REV(LIST(SR,NH(FH))) BY AX04,AX3
              .=REV(LIST(SR,NH(FH)))+NILL(PTR(LREC)) BY CONC1
              .=REV(LIST(SR,NH(FH)))+LIST(TR,NH(FH)) BY T1
              .=LIST(TR,NH(H')) BY AX13,A1;

    LISTPROGPROPERTY:
       EX H ST CP(H)=FINISH & LIST(TR,H)=REV(LIST(SR,FH))
                                                  BY C1,C2
  END

THANKS,O.K
----------
```