Erratum

Proving Properties of Pascal Programs in MIZAR 2

Piotr Rudnicki and Wlodzimierz 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

```
we obtain
```

![Fig. 1](image1)

![Fig. 2](image2)
2. For a cyclic list

![Diagram](image1)

Fig. 3

we obtain

![Diagram](image2)

Fig. 4

3. For a cyclic list with a "start"

![Diagram](image3)

Fig. 5

the program also terminates, and in this case we obtain

![Diagram](image4)

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 a diagonal part of a MIZAR 2 text is called an environment and contains definitions and proofs. The lines following the heading 'Program description' are presented to be processed by a describer. The lines following the heading 'Description technology' are prepared once and by hand, and later on are appended (possibly by persons) to a particular problem. The fragments of environments pertinent to a particular domain, in our case to arithmetic and set theory, are to be prepared by hand (but 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 OK is the checker signal confirming correctness of the proof.
The proof for the factorial computing program in MISAR 2

ENVIRON

**DESCRIPTION TECHNOLOGY**

**TYPE HISTORY**
- LET M+M DENOTE HISTORY
- GIVEN FM BEING HISTORY
- FOR M CHECK FM BEING HISTORY
- CONSIDER FM BEING NATURAL
- FOR M BEING NATURAL RECONSIDER M AS ELEMENT OF INTEGERS
- FOR N+L N DENOTE ELEMENT OF INTEGERS
- FOR N+L RECONSIDER N+L AS ELEMENT OF INTEGERS
- FOR N+L PRED K (= N)
- FOR A BEING NONEMPTY CONSIDER A RESTING NONEMPTY
- DEFINITION LET A BE NONEMPTY, V BE ELEMENT OF P(T(A)), 
  E BE HISTORY, C BE ELEMENT OF A
- PRED C = VAL(V, A) ENDI

**ARITHMETICS**

**SCHEME INDUCTION**

**PROOF PI**

**END**

**REMARKS**

**CONDITIONS**

**FOR N ST = 0 = N HOLDS FCN**

**SINCE**

**COND1 FCN=3**

**COND2 FOR N ST = (N+1) FCN = (FCN+1)**

**FOR K CONSIDER FACTORIAL BEING ELEMENT OF INTEGERS**

**AX1 FACTORIAL(0) = 1**

**AX2 FOR N ST = (N+1) = K HOLDS (K * (N+1))**

**AX3 FOR N ST = 0 = N HOLDS (N+1)**

**AX4 FOR N ST = (N+1) HOLDS FACTORIAL(N) * (N+1) = FACTORIAL(N+1)**

**PROGRAM DESCRIPTION**

**LET V DENOTE ELEMENT OF P(T(INTEGERS))**

**CONSIDER X+Y Z BEING ELEMENT OF P(T(INTEGERS))**

**AX01 CFN(NZ(F)) = I**

**AX02 VAL(NZ(F)) = 1**

**AX03 VAL(NZ(F)) = VAL(NZ(F))**

**LAMBDA (X Z = 0)**

**AX11 CFN(NZ(F)) = 1**

**AX12 VAL(NZ(F)) = 0**

**AX13 VAL(NZ(F)) = 1**

**WHILE Z X BEGIN**

**AX21 FOR N ST CF(N) = 2 HOLES**

**LAMBDA (Z=N) VAL(W, A) = CF(N+1)**

**AX22 FOR N ST CF(N) = 2 HOLDS VAL(W, A) = FINISH**

**AX23 HOLES CF(N)=2**

**AX24 FOR N ST CF(N)=2 HOLES VAL(W, A) = VAL(W, A)**

**AX25 FOR N ST CF(N)=A HOLES VAL(W, A) = VAL(W, A)**

**AX26 FOR N ST CF(N)=A HOLES VAL(W, A) = VAL(W, A)**

**END**
SAT3 FOR x:ST OP(H) = A & y:Y HOLDS VAL(y,H(1H)) = VAL(x(1H))

-- ( *FINISH*) END-

AY1 Y1 I Z12

AXVAL1 FOR M HOLDS VAL (x,H) = VAL(x(1H))

-- PROPERTIES --------------------------------------------------

BEGIN

FIRSTCONDITION

AX1

ASSUME a16 (= VAL(x,FH))

VAL(y, x(x(MH)(1H))) = FACTORIAL(y) BY AX1-AX13

HENCE EX x:ST OP(H) = 2 & VAL(x,H) = FACTORIAL(y)

VAL(y,H) = FACTORIAL(y) BY AX11-AX12

END

SECONDCONDITION

AX12

LET R BE ELEMENT OF INTEGERS SUCH THAT

A00 (+ = N XOR

A11 N (= VAL(x,FH)) IMPLIES

IX X ST OP(H) = 2 & VAL(x,H) = FACTORIAL(H)

ASSUME A21 N(+1) (= VAL(x,FH)) THUS EX x:ST OP(H) = 2 & VAL(x,H) = FACTORIAL(H)

PROOF

x (= VAL(x,FH)) BY AX2-A21

THEN CONSIDER M SUCH THAT

A33 OP(H) = 2 & VAL(x,H) = VAL(y, H) = FACTORIAL(H) BY AX1

N (= VAL(y,FH)) BY AX2-A21

THEN VAL(x,H) = VAL(x,y) BY AXVAL2

THEN CP11 CP(H) = z BY AX2-A13

THEN CP3 CP(H) = z BY AX1

Z11 VAL(x,y) = VAL(z) BY AX2-A21

Z31 VAL(x,H) = VAL(z,H) BY AX2-A21

Z19 VAL(x,H) = z BY AX2-A21

Z23 VAL(x,H) = z BY AX2-A21

Z31 VAL(x,H) = z BY AX2-A21

Z19 VAL(x,H) = z BY AX2-A21

HENCE EX x:ST OP(H) = 2 & VAL(x,H) = N(+1) & VAL(x,H) = FACTORIAL(H)

END --- SECONDSTEP

Lemma

FOR M ST OP(H) = A & N (= VAL(x,FH)) IMPLIES

IX X ST CP(H) = A & VAL(x,H) = A & VAL(x,H) = FACTORIAL(H)

FROM INDUCTION(FIRSTCONDITION,SECONDCONDITION)

FACTFROMPROPERTY

z (= VAL(x,FH)) IMPLIES

EX x:ST OP(H) = A & VAL(x,H) = FACTORIAL(z)

PROOF

ASSUME A11 t (= VAL(x,FH))

VAL(x,FH) (= VAL(x,FH) BY AX2)
Appendix 2

The proof for the list reversal program in MJAX 2

---

**DESCRIPTION TECHNOLOGY**

TYPE HISTORY LET W Nathaniel DEMO HISTORY.

GIVEN Pk HISTORICAL.

FOR K CONSIDER AT BEING HISTORICAL.

CONSIDER FINISH BEING HISTORICAL.

LET A DENOTE NONEMPTY.

FOR A CONSIDER ALL BEING ELEMENT OF PIR(A).

DEFINITION LET A BE NONEMPTY = BE ELEMENT OF PIR(A).

PRED Cworthy OR END.

DEFINITION LET A BE NONEMPTY = BE ELEMENT OF PIR(A).

PRED YFTed(NAV,NUM) END.

---

**LIST THEORY**

TYPE LIST OF A BEING NONEMPTY.

FOR A CONSIDER NULL BEING LIST OF A.

FOR A BEING NONEMPTY E BEING ELEMENT OF A.

L BEING LIST OF A.

RECURSIVE E L AS LIST OF A.

LIST IF A BEING NONEMPTY.

L BEING LIST OF A.

VALUES E.(L) NULL(A).

SCHEME LISTINGS CONSIDER A BEING NONEMPTY.

PRED P1.

FOR L BEING LIST OF A BEING FILL.

SINCE

CONS FOR L BEING ELEMENT OF A.

L BEING LIST OF A.

VALUES FILL(L).

END.

FOR A BEING NONEMPTY.

L L BEING LIST OF A.

RECOGNIZED L L AS LIST OF A.

CONS FOR A BEING NONEMPTY.

L BEING LIST OF A.

VALUES NULL(L) NULL(L).

DEFINITION LET A BE NONEMPTY.

L L BEING LIST OF A.

PRED L REV(L) END.

REKV1 FOR A BEING NONEMPTY.

VALUES REV(NILL(A)) NILL(A).

---

THANKS TO:

---
REXLMA

FOR A BEING X WHERE Y IS BEING ELEMENT OF A)
L. L' BEING LIST OF A
HOLDS NEXT(L, L') = NEXT(L')

DEFINITION LET A BE BEING ELEMENT OF A
L. L' BEING LIST OF A
PREL E IS MEMBER OF L END

FOR A BEING MEMPRED L (DENOTE NOT (EX X BEING ELEMENT OF A ST X IS MEMBER OF L A & A 7 MEMBER OF L'))
AKLXLY FOR A BEING MEMPRED L, L' BEING LIST OF A
ST NIL(VAL) HOLDS L' MEMPRED L'

**PROGRAM DESCRIPTION**

TYPE UNREC-RECORD NEXT * UNREC CONT INTEGER END
GIVEN RECORD BEING MEMPRED
FOR V BEING ELEMENT IF PTR(RELU)
TIME NEXT(*ID(LE(U), V), P(LE(U)))
CONT(FLG(LE(U), V), V, INTEGER)

**LIST REPRESENTATION**

DEFINITION LET V BE ELEMENT IF PTR(RELU)
L. L' BE LIST OF PTR(RELU) X IN HISTORY
PREL LIST(L, V) EVER

LET V BE ELEMENT OF PTR(RELU)
AKLXLY FOR A BEING MEMPRED L, L' BE LIST OF PTR(RELU) X
REP2 FOR W. V V HOLDS L. L' LIST(W, V)

REP3 FOR W. V ST VAL(W) = VAL(V)
AKLXLY FOR A BEING MEMPRED L, L' BE LIST OF PTR(RELU) X

AKLXLY FOR W. V V NOT HOLDS L. L' LIST(W, V) X

THILX FOR L, L', X. ST ELL MEMPRED L' HOLDS L MEMPRED ELL

**PROGRAM DESCRIPTION CONTINUED**


CONSIDER LAUX-SR-TR BEING ELEMENT OF PTR(RELU)
AKLXLY FOR W. V ST V BEING ELEMENT OF PW(RELU)

AK41 FOR W. V ST WLHOLDS L. L' HOLDS(W, V)

AK42 FOR W. V ST WRHOLDS L. L' NOT HOLDS(W, V)

AK43 FOR W. V ST V BEING ELEMENT OF PW(RELU)

AK44 FOR W. V ST V BEING ELEMENT OF PW(RELU)

AK45 FOR W. V ST V BEING ELEMENT OF PW(RELU)

AK46 FOR H. W SR-TR H. W HOLDS(RELU)

AK47 FOR H. W SR-TR H. W NOT HOLDS(RELU)

AK48 FOR H. W SR-TR H. W NOT HOLDS(RELU)

AK49 FOR H. W SR-TR H. W NOT HOLDS(RELU)

AK50 FOR H. W SR-TR H. W NOT HOLDS(RELU)

AK51 FOR H. W SR-TR H. W NOT HOLDS(RELU)

AK52 FOR H. W SR-TR H. W NOT HOLDS(RELU)

AK53 FOR H. W SR-TR H. W NOT HOLDS(RELU)

AX24 FOR G. H. ST CP(H) = CP(G) & VAL(AUX) = VAL(LE(H))

AX23 FOR G. H. ST CP(H) = CP(G) & VAL(AUX) = VAL(LE(H))

AX22 FOR G. H. ST CP(H) = CP(G) & VAL(AUX) = VAL(LE(H))

AX21 FOR G. H. ST CP(H) = CP(G) & VAL(AUX) = VAL(LE(H))
AX251 FOR H ST CP(H)="1"
    HOLD(LIST(LAUX(H))=LIST(NEXT(VM(SH(H)))=H)
    ****CP 3M=SW-NEXT=1ST
    AX31 FOR H ST CP(H)=2 HOLD(S CP(H))=H
    AX32 FOR H ST CP(H)=2
    HOLD(NEXT(VM(SH(H)))=VM(SH(H))=H)
    AX33 FOR H ST CP(H)=3 & NOT VM(SH(H)) 15 MEMBER OF LIST(TRM)
    HOLD(LIST(NEXT(VM(SH(H)))=H)="LIST(TRM)"
    AX34 FOR H=V
    ST CP(H)=3 & NOT VM(SH(H)) MEMBERS OF LIST(TRM)
    HOLD(LIST(NEXT(VM(SH(H)))="LIST(TRM)"
    ****(CP 4V) TV=R2I
    AX41 FOR H ST CP(H)=H HOLD(S CP(H))=H
    AX42 FOR H ST CP(H)=H HOLD(VM(SH(H))=VM(SH(H))=H)
    AX43 FOR H ST CP(H)=H HOLD(VM(SH(H))=VM(SH(H))=H)
    ST CP(H)=H & VM(SH(H))=VM(SH(H))=H
    AX44 FOR H ST CP(H)=H HOLD(List(TRM(H)))="LIST(TRM(H))"
    AX45 FOR H ST CP(H)=H & VM(SH(H))=H
    HOLD(List(VM(H))="LIST(VM(H))"
    ****(CP 5V) SR=V
    AX51 FOR H ST CP(H)=H HOLD(S CP(H))=H
    AX52 FOR H ST CP(H)=H HOLD(S VM(SH(H))=VM(SH(H))=H)
    AX53 FOR A BEING MEMBERS, VM(SH(H))=H
    ST CP(H)=H & VM(SH(H))=VM(SH(H))=H
    AX54 FOR H ST CP(H)=H HOLD(List(SH(H)))="LIST(SH(H))"
    AX55 FOR H ST CP(H)=H & VM(SH(H))=H
    HOLD(List(VM(H))="LIST(VM(H))"
    **** EMK=(CP 5V) 5V=(H)
BEGIN
    FINISH
NOW LET N BE HISTORY SUCH THAT
A1 CP(NX(H)) = null PTR(NX(H))=N LIST(NK(H)) &
   NH=PR(NX(H)) TIMES LIST(NK(H))=N
LIST(NK(H))=null PTR(NX(H))=N LIST(NK(H))=null
HERE EX H ST CP(H)=1 & VM(SH(H))=VM(SH(H))=H
   LIST(VM(H))=null PTR(NX(H))=N LIST(TRM(H))=null
   BY A151
END **FIRST**

BEGIN
NOW LET C IN (ELEVEN OF PTR(NX(H)))
   I BE LIST OF PINS(REC) SUCH THAT
A11 FOR H BEING HISTORY SUCH THAT
ST CP(H)=1 & VM(SH(H))=I & L=MEMBERS LIST(TRM(H)
EX H ST CP(H)=1 & VM(SH(H))=I & L=MEMBERS LIST(TRM(H)
   TIMES EX H ST CP(H)=1 & VM(SH(H))=I & L=MEMBERS
   LIST(TRM(H))=null
   LIST(REC(H)=null PTR(NX(H))=N LIST(VM(H))=null
   L=MEMBERS LIST(TRM(H))=null
   LIST(REC(H)=null PTR(NX(H))=N LIST(VM(H))=null
   L=MEMBERS
   LIST(REC(H)=null PTR(NX(H))=N LIST(VM(H))=null
   PROOF
   L=MEMBERS
   LIST(REC(H)=null PTR(NX(H))=N LIST(VM(H))=null
   THEN LIST(BR(H))=null PTR(REC(H))=N A21
   THEN VM(SH(H))=null(MEMBER L=MEMBERS
   LIST(REC(H)=null PTR(NX(H))=N LIST(VM(H))=null
   PROOF
   L=MEMBERS
   LIST(REC(H)=null PTR(NX(H))=N LIST(VM(H))=null
   THEN CP(H)=null PTR(REC(H))=N A21
   THEN CP(H)=null PTR(REC(H))=N A21
   THEN CP(H)=null PTR(REC(H))=N A21
   THEN CP(H)=null PTR(REC(H))=N A21
   CP(H)=null PTR(REC(H))=N A21
   THEN CP(H)=null PTR(REC(H))=N A21
TI: AXI1: (PTR(LREC)) = LIST(TRM(H(FH)) BY AXI2:REFS11
SI: LIST(SR+NH(FH)) = LIST(SR+NH(FH)) BY TI:AXI1:II
LIST(SR+NH(FH)) = LIST(SR+NH(FH)) BY REFLEXIVE: AXCL
THEN CONSIDER IF BEING HISTORY SUCH THAT
ALL CP(OP1) = &NH(CR1) = NH(LREC) A
LIST(TM') = REV(LIST(SR+NH(FH))) = LIST(TRM(H(FH))
BY LOOP:AXI1:SI1
CI: CP(NH(MH)) = FINISH BY AXI1:SI1
C2: REV(LIST(SR+NH(FH))) = REV(LIST(SR+NH(FH))) BY AXI4:AXS
+REV(LIST(SR+NH(FH))) = NILL (PTR(LREC)) BY CONC
+REV(LIST(SR+NH(FH))) = LIST(SR+NH(FH)) BY T1
+LIST(TM'(MH')) BY AXI3:SI1
LISTPROP: EX H ST CP(H) = FINISH & LIST(TM) = REV(LIST(SR+FH))
BY CI:C2

THANKS D.K.