```
#|

 Copyright (C) 1994 by Robert S. Boyer and J Strother Moore.  All Rights
 Reserved.

 This script is hereby placed in the public domain, and therefore unlimited
 editing and redistribution is permitted.

 NO WARRANTY

 Robert S. Boyer and J Strother Moore PROVIDE ABSOLUTELY NO WARRANTY.  THE
 EVENT SCRIPT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS
 OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF
 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE RISK AS TO
 THE QUALITY AND PERFORMANCE OF THE SCRIPT IS WITH YOU.  SHOULD THE SCRIPT
 PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR
 CORRECTION.

 IN NO EVENT WILL Robert S. Boyer or J Strother Moore BE LIABLE TO YOU FOR ANY
 DAMAGES, ANY LOST PROFITS, LOST MONIES, OR OTHER SPECIAL, INCIDENTAL OR
 CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THIS SCRIPT
 (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE
 OR LOSSES SUSTAINED BY THIRD PARTIES), EVEN IF YOU HAVE ADVISED US OF THE
 POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

|#

; This is the list of verification conditions for a version of our
; majority vote algorithm.  For the details of the algorithm, see the
; comment at the end of the file.  Boyer and Moore.

; This list of events has been further edited, for processing by
; DO-FILE, by (1) inserting the following NOTE-LIB, (2) commenting out
; each FORTRAN-COMMENT and following the comment with the
; corresponding macroexpansion, and (3) by commenting out each
; (COMMENT ...).
```

EVENT: Start with the library "fortran" using the compiled version.

```
; (FORTRAN-COMMENT FORTRAN)
```

AXIOM: fortran
t

EVENT: Introduce the function symbol *a$0* of 0 arguments.

EVENT: Introduce the function symbol *boole$0* of 0 arguments.

EVENT: Introduce the function symbol *boole$1* of 0 arguments.

EVENT: Introduce the function symbol *boole$2* of 0 arguments.

EVENT: Introduce the function symbol *boole$3* of 0 arguments.

EVENT: Introduce the function symbol *boole$4* of 0 arguments.

EVENT: Introduce the function symbol *boole$5* of 0 arguments.

EVENT: Introduce the function symbol *boole$6* of 0 arguments.

EVENT: Introduce the function symbol *boole$7* of 0 arguments.

EVENT: Introduce the function symbol *boole$8* of 0 arguments.

EVENT: Introduce the function symbol *cand$0* of 0 arguments.

EVENT: Introduce the function symbol *cand$1* of 0 arguments.

EVENT: Introduce the function symbol *cand$2* of 0 arguments.

EVENT: Introduce the function symbol *cand$3* of 0 arguments.

EVENT: Introduce the function symbol *cand$4* of 0 arguments.

EVENT: Introduce the function symbol *cand$5* of 0 arguments.

EVENT: Introduce the function symbol *cand$6* of 0 arguments.

EVENT: Introduce the function symbol *cand$7* of 0 arguments.

EVENT: Introduce the function symbol *cand$8* of 0 arguments.

EVENT: Introduce the function symbol *i$0* of 0 arguments.

EVENT: Introduce the function symbol *i$1* of 0 arguments.

EVENT: Introduce the function symbol *i$2* of 0 arguments.

EVENT: Introduce the function symbol *i$3* of 0 arguments.

EVENT: Introduce the function symbol *i$4* of 0 arguments.

EVENT: Introduce the function symbol *i$5* of 0 arguments.

EVENT: Introduce the function symbol *i$6* of 0 arguments.

EVENT: Introduce the function symbol *i$7* of 0 arguments.

EVENT: Introduce the function symbol *i$8* of 0 arguments.

EVENT: Introduce the function symbol *k$1* of 0 arguments.

EVENT: Introduce the function symbol *k$2* of 0 arguments.

EVENT: Introduce the function symbol *k$3* of 0 arguments.

EVENT: Introduce the function symbol *k$4* of 0 arguments.

EVENT: Introduce the function symbol *k$5* of 0 arguments.

EVENT: Introduce the function symbol *k$6* of 0 arguments.

EVENT: Introduce the function symbol *k$7* of 0 arguments.

EVENT: Introduce the function symbol *k$8* of 0 arguments.

EVENT: Introduce the function symbol *n$0* of 0 arguments.

DEFINITION:
cnt $(x,\ a,\ i,\ n)$
$=$ **if** $(n \simeq \mathtt{0}) \vee (n < i)$ **then** $\mathtt{0}$
elseif zeqp $(x,$ elt1 $(a,\ n))$ **then** $1 +$ cnt $(x,\ a,\ i,\ n - 1)$
else cnt $(x,\ a,\ i,\ n - 1)$ **endif**

AXIOM: input-conditions
$((\mathtt{'0} < j) \wedge ((\text{N\$0} \not< j) \wedge (\neg$ negativep $(\text{elt1}\,(a,\ j)))))$
$\rightarrow$ $(\text{elt1}\,(a,\ j) \in \mathbf{N})$

DEFINITION:
GLOBAL-HYPS
$=$ $((\text{N\$0} \neq \mathtt{'0})$
$\wedge$ $(((1 + \text{N\$0}) <$ LEAST-INEXPRESSIBLE-POSITIVE-INTEGER$)$
$\wedge$ $(\text{N\$0} \in \mathbf{N})))$

THEOREM: plus-1
$(\mathtt{1} + x) = (1 + x)$

THEOREM: difference-0
$(x \not< y) \rightarrow ((y - x) = \mathtt{0})$

THEOREM: difference-1
$(x - \mathtt{1}) = (x - 1)$

THEOREM: lessp-x-1
$(x < \mathtt{1}) = (x \simeq \mathtt{0})$

THEOREM: lessp-remainder2
$((x \bmod y) < y) = (y \not\simeq \mathtt{0})$

THEOREM: remainder-quotient-elim
$((y \not\simeq \mathtt{0}) \wedge (x \in \mathbf{N})) \rightarrow (((x \bmod y) + (y * (x \div y))) = x)$

THEOREM: quotient-by-2-bound
$x \not< (x \div \mathtt{2})$

THEOREM: lessp-quotient-rewrite
$((n \div \mathtt{2}) < m) = (n < (m + m))$

THEOREM: znormalize-zero
$(x \in \mathbf{N}) \rightarrow ((\text{znormalize}\,(x) = \mathtt{0}) = (x = \mathtt{0}))$

EVENT: Disable znormalize.

THEOREM: cnt-bound
$n \not< \text{cnt}\,(x,\ a,\ \texttt{1},\ n)$

THEOREM: input-definedness
GLOBAL-HYPS $\rightarrow$ `'*1*true`

```
#| (COMMENT) |#

; (FORTRAN-COMMENT INPUT)
```

AXIOM: input
t

AXIOM: assignment
(BOOLE$1 = BOOLE$0)
$\wedge$ ((CAND$1 = CAND$0) $\wedge$ ((I$1 = I$0) $\wedge$ (K$1 = '0)))

```
; (FORTRAN-COMMENT LOGICAL-IF-F)
```

AXIOM: logical-if-f
t

AXIOM: assignment1
(BOOLE$2 = BOOLE$1)
$\wedge$ ((CAND$2 = CAND$1) $\wedge$ ((I$2 = '1) $\wedge$ (K$2 = K$1)))

```
; (FORTRAN-COMMENT LOGICAL-IF-T)
```

AXIOM: logical-if-t
t

AXIOM: effects-of-undefiner
(BOOLE$3 = BOOLE$2) $\wedge$ ((CAND$3 = CAND$2) $\wedge$ (K$3 = K$2))

```
; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

AXIOM: assignment2
(BOOLE$4 = `'*1*false`)
$\wedge$ ((CAND$4 = CAND$3) $\wedge$ ((I$4 = I$3) $\wedge$ (K$4 = K$3)))

THEOREM: output
(zeqp (K$3, '0) $\wedge$ (I$2 $\wedge$ (zgreaterp (I$2, N$0) $\wedge$ GLOBAL-HYPS)))
$\rightarrow$ (((BOOLE$4 = `'*1*true`) $\vee$ (BOOLE$4 = `'*1*false`))
$\wedge$ **if** BOOLE$4
then znumberp (CAND$4)
$\wedge$ ((N$0 $\div$ '2) < cnt (CAND$4, A$0, '1, N$0))
else (N$0 $\div$ '2) $\not<$ cnt (x, A$0, '1, N$0) **endif**)

```
#| (COMMENT INPUT F T T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F1)
```

AXIOM: logical-if-f1
t

AXIOM: assignment2
(BOOLE$4 = '*1*true)
∧ ((CAND$4 = CAND$3) ∧ ((I$4 = I$3) ∧ (K$4 = K$3)))

THEOREM: input-cond-of-zquotient
((¬ zeqp (K$3, '0)) ∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ GLOBAL-HYPS)))
→ ((¬ zeqp ('2, '0)) ∧ expressible-znumberp (zquotient (N$0, '2)))

```
#| (COMMENT INPUT F T F) |#
```

```
; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

THEOREM: output
(zgreaterp (K$4, zquotient (N$0, '2))
∧ ((¬ zeqp (K$3, '0))
∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ GLOBAL-HYPS))))
→ (((BOOLE$4 = '*1*true) ∨ (BOOLE$4 = '*1*false))
∧ **if** BOOLE$4
then znumberp (CAND$4)
∧ ((N$0 ÷ '2) < cnt (CAND$4, A$0, '1, N$0))
else (N$0 ÷ '2) ≮ cnt (*x*, A$0, '1, N$0) **endif**)

```
#| (COMMENT INPUT F T F T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F2)
```

AXIOM: logical-if-f2
t

AXIOM: assignment3
(BOOLE$5 = BOOLE$4)
∧ ((CAND$5 = CAND$4) ∧ ((I$5 = I$4) ∧ (K$5 = '0)))

```
; (FORTRAN-COMMENT LOGICAL-IF-F3)
```

AXIOM: logical-if-f3
t

AXIOM: assignment4
(BOOLE$6 = BOOLE$5)
∧ ((CAND$6 = CAND$5) ∧ ((I$6 = '1) ∧ (K$6 = K$5)))

```
; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

AXIOM: effects-of-undefiner1
(BOOLE$7 = BOOLE$6) ∧ ((CAND$7 = CAND$6) ∧ (K$7 = K$6))

AXIOM: assignment5
(BOOLE$8 = '*1*false)
∧ ((CAND$8 = CAND$7) ∧ ((I$8 = I$7) ∧ (K$8 = K$7)))

THEOREM: output
(I$6
∧ (zgreaterp (I$6, N$0)
∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
∧ ((¬ zeqp (K$3, '0))
∧ (I$2
∧ (zgreaterp (I$2, N$0) ∧ GLOBAL-HYPS))))))
→ (((BOOLE$8 = '*1*true) ∨ (BOOLE$8 = '*1*false))
∧ **if** BOOLE$8
then znumberp (CAND$8)
∧ ((N$0 ÷ '2) < cnt (CAND$8, A$0, '1, N$0))
else (N$0 ÷ '2) ≮ cnt ($x$, A$0, '1, N$0) **endif**)

```
#| (COMMENT INPUT F T F F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F4)
```

AXIOM: logical-if-f4
t

THEOREM: array-bounds-check-for-a
((¬ zgreaterp (I$6, N$0))
 ∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
  ∧ ((¬ zeqp (K$3, '0))
   ∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ GLOBAL-HYPS)))))
→ (('0 < I$6) ∧ (N$0 ≮ I$6))

```
#| (COMMENT INPUT F T F F F F) |#
```

THEOREM: definedness
((¬ zgreaterp (I$6, N$0))
 ∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
  ∧ ((¬ zeqp (K$3, '0))
   ∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ GLOBAL-HYPS)))))
→ znumberp (CAND$6)

```
#| (COMMENT INPUT F T F F F F) |#
```

THEOREM: definedness1
((¬ zgreaterp (I$6, N$0))
 ∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
  ∧ ((¬ zeqp (K$3, '0))
   ∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ GLOBAL-HYPS)))))
→ znumberp (elt1 (A$0, I$6))

```
#| (COMMENT INPUT F T F F F F) |#

; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

THEOREM: phase2-invrt
(zneqp (CAND$6, elt1 (A$0, I$6))
 ∧ ((¬ zgreaterp (I$6, N$0))
  ∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
   ∧ ((¬ zeqp (K$3, '0))
    ∧ (I$2
     ∧ (zgreaterp (I$2, N$0) ∧ GLOBAL-HYPS))))))
→ (((I$6 ∈ **N**)

$$
\begin{array}{l}
\wedge \quad ((\textsc{i}\$6 \neq \texttt{'0}) \\
\quad \wedge \quad ((\textsc{boole}\$6 = \texttt{'*1*true}) \\
\quad\quad \wedge \quad ((\textsc{n}\$0 \not< \textsc{i}\$6) \\
\quad\quad\quad \wedge \quad (((\neg\ \text{negativep}\,(\textsc{cand}\$6)) \\
\quad\quad\quad\quad \rightarrow \quad (\textsc{cand}\$6 \in \mathbf{N})) \\
\quad\quad\quad\quad \wedge \quad ((\text{zneqp}\,(x,\ \textsc{cand}\$6) \\
\quad\quad\quad\quad\quad \rightarrow \quad (\textsc{n}\$0 \\
\quad\quad\quad\quad\quad\quad \not< \quad (\texttt{'2} \\
\quad\quad\quad\quad\quad\quad\quad * \quad \text{cnt}\,(x, \\
\quad\quad\quad\quad\quad\quad\quad\quad \textsc{a}\$0, \\
\quad\quad\quad\quad\quad\quad\quad\quad \texttt{'1}, \\
\quad\quad\quad\quad\quad\quad\quad\quad \textsc{n}\$0)))) \\
\quad\quad\quad\quad\quad \wedge \quad ((\textsc{n}\$0 \\
\quad\quad\quad\quad\quad\quad \not< \quad (\texttt{'2} \\
\quad\quad\quad\quad\quad\quad\quad * \quad \text{cnt}\,(\textsc{cand}\$6, \\
\quad\quad\quad\quad\quad\quad\quad\quad \textsc{a}\$0, \\
\quad\quad\quad\quad\quad\quad\quad\quad \texttt{'1}, \\
\quad\quad\quad\quad\quad\quad\quad\quad \textsc{i}\$6))) \\
\quad\quad\quad\quad\quad\quad \wedge \quad (\textsc{k}\$6 \\
\quad\quad\quad\quad\quad\quad\quad = \quad \text{cnt}\,(\textsc{cand}\$6, \\
\quad\quad\quad\quad\quad\quad\quad\quad \textsc{a}\$0, \\
\quad\quad\quad\quad\quad\quad\quad\quad \texttt{'1}, \\
\quad\quad\quad\quad\quad\quad\quad\quad \textsc{i}\$6))))))))) \\
\quad \wedge \quad \text{lex}\,(\text{cons}\,((1 + \textsc{n}\$0) - \textsc{i}\$6,\ \texttt{'nil}), \\
\quad\quad\quad \text{cons}\,(1 + (1 + (\textsc{n}\$0 + \textsc{n}\$0)),\ \texttt{'nil})))
\end{array}
$$

```
#| (COMMENT INPUT F T F F F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F5)
```

AXIOM: logical-if-f5
t

THEOREM: input-cond-of-zplus

$$
\begin{array}{l}
((\neg\ \text{zneqp}\,(\textsc{cand}\$6,\ \text{elt1}\,(\textsc{a}\$0,\ \textsc{i}\$6))) \\
\wedge \quad ((\neg\ \text{zgreaterp}\,(\textsc{i}\$6,\ \textsc{n}\$0)) \\
\quad\quad \wedge \quad ((\neg\ \text{zgreaterp}\,(\textsc{k}\$4,\ \text{zquotient}\,(\textsc{n}\$0,\ \texttt{'2}))) \\
\quad\quad\quad \wedge \quad ((\neg\ \text{zeqp}\,(\textsc{k}\$3,\ \texttt{'0})) \\
\quad\quad\quad\quad \wedge \quad (\textsc{i}\$2 \\
\quad\quad\quad\quad\quad \wedge \quad (\text{zgreaterp}\,(\textsc{i}\$2,\ \textsc{n}\$0) \wedge \textsc{global-hyps})))))) \\
\rightarrow \quad \text{expressible-znumberp}\,(\text{zplus}\,(\textsc{k}\$6,\ \texttt{'1}))
\end{array}
$$

```
#| (COMMENT INPUT F T F F F F F) |#
```

AXIOM: assignment5
(BOOLE$7 = BOOLE$6)
∧ ((CAND$7 = CAND$6) ∧ ((I$7 = I$6) ∧ (K$7 = zplus (K$6, '1'))))

THEOREM: input-cond-of-zquotient1
((¬ zneqp (CAND$6, elt1 (A$0, I$6)))
∧ ((¬ zgreaterp (I$6, N$0))
∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2')))
∧ ((¬ zeqp (K$3, '0'))
∧ (I$2
∧ (zgreaterp (I$2, N$0) ∧ GLOBAL-HYPS))))))
→ ((¬ zeqp ('2, '0')) ∧ expressible-znumberp (zquotient (N$0, '2')))

```
#| (COMMENT INPUT F T F F F F F) |#

; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

THEOREM: output
(zgreaterp (K$7, zquotient (N$0, '2'))
∧ ((¬ zneqp (CAND$6, elt1 (A$0, I$6)))
∧ ((¬ zgreaterp (I$6, N$0))
∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2')))
∧ ((¬ zeqp (K$3, '0'))
∧ (I$2
∧ (zgreaterp (I$2, N$0)
∧ GLOBAL-HYPS)))))))
→ (((BOOLE$7 = '*1*true') ∨ (BOOLE$7 = '*1*false'))
∧ **if** BOOLE$7
then znumberp (CAND$7)
∧ ((N$0 ÷ '2) < cnt (CAND$7, A$0, '1, N$0))
else (N$0 ÷ '2) ≮ cnt (*x*, A$0, '1, N$0) **endif**)

```
#| (COMMENT INPUT F T F F F F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F6)
```

AXIOM: logical-if-f6
t

THEOREM: phase2-invrt

$$
\begin{array}{l}
((\neg\ \mathrm{zgreaterp}\,(\mathrm{K}\$7,\ \mathrm{zquotient}\,(\mathrm{N}\$0,\ \texttt{'2}))) \\
\ \wedge\quad ((\neg\ \mathrm{zneqp}\,(\mathrm{CAND}\$6,\ \mathrm{elt1}\,(\mathrm{A}\$0,\ \mathrm{I}\$6))) \\
\qquad \wedge\quad ((\neg\ \mathrm{zgreaterp}\,(\mathrm{I}\$6,\ \mathrm{N}\$0)) \\
\qquad\quad \wedge\quad ((\neg\ \mathrm{zgreaterp}\,(\mathrm{K}\$4,\ \mathrm{zquotient}\,(\mathrm{N}\$0,\ \texttt{'2}))) \\
\qquad\qquad \wedge\quad ((\neg\ \mathrm{zeqp}\,(\mathrm{K}\$3,\ \texttt{'0})) \\
\qquad\qquad\quad \wedge\quad (\mathrm{I}\$2 \\
\qquad\qquad\qquad \wedge\quad (\mathrm{zgreaterp}\,(\mathrm{I}\$2,\ \mathrm{N}\$0) \\
\qquad\qquad\qquad\quad \wedge\quad \mathrm{GLOBAL\text{-}HYPS}))))))) \\
\rightarrow\quad (((\mathrm{I}\$7 \in \mathbf{N}) \\
\qquad \wedge\quad ((\mathrm{I}\$7 \not\equiv \texttt{'0}) \\
\qquad\quad \wedge\quad ((\mathrm{BOOLE}\$7 = \texttt{'*1*true}) \\
\qquad\qquad \wedge\quad ((\mathrm{N}\$0 \not< \mathrm{I}\$7) \\
\qquad\qquad\quad \wedge\quad (((\neg\ \mathrm{negativep}\,(\mathrm{CAND}\$7)) \\
\qquad\qquad\qquad \rightarrow\quad (\mathrm{CAND}\$7 \in \mathbf{N})) \\
\qquad\qquad\qquad \wedge\quad ((\mathrm{zneqp}\,(x,\ \mathrm{CAND}\$7) \\
\qquad\qquad\qquad\qquad \rightarrow\quad (\mathrm{N}\$0 \\
\qquad\qquad\qquad\qquad\qquad \not< \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad\qquad\qquad * \quad \mathrm{cnt}\,(x, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{A}\$0, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{N}\$0)))) \\
\qquad\qquad\qquad\qquad \wedge\quad ((\mathrm{N}\$0 \\
\qquad\qquad\qquad\qquad\qquad \not< \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad\qquad\qquad * \quad \mathrm{cnt}\,(\mathrm{CAND}\$7, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{A}\$0, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{I}\$7))) \\
\qquad\qquad\qquad\qquad\qquad \wedge\quad (\mathrm{K}\$7 \\
\qquad\qquad\qquad\qquad\qquad\qquad = \quad \mathrm{cnt}\,(\mathrm{CAND}\$7, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{A}\$0, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{I}\$7))))))))) \\
\qquad \wedge\quad \mathrm{lex}\,(\mathrm{cons}\,((1 + \mathrm{N}\$0) - \mathrm{I}\$7,\ \texttt{'nil}), \\
\qquad\qquad\quad \mathrm{cons}\,(1 + (1 + (\mathrm{N}\$0 + \mathrm{N}\$0)),\ \texttt{'nil})))
\end{array}
$$

```
#| (COMMENT INPUT F T F F F F F F) |#
```

EVENT: Undo back through the event named 'logical-if-t'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F1)
```

AXIOM: logical-if-f1
t

```
; (FORTRAN-COMMENT LOGICAL-IF-T)
```

AXIOM: logical-if-t
t

THEOREM: array-bounds-check-for-a
(zeqp (K$2, '0) ∧ ((¬ zgreaterp (I$2, N$0)) ∧ GLOBAL-HYPS))
→ (('0 < I$2) ∧ (N$0 ≮ I$2))

```
#| (COMMENT INPUT F F T) |#
```

THEOREM: definedness
(zeqp (K$2, '0) ∧ ((¬ zgreaterp (I$2, N$0)) ∧ GLOBAL-HYPS))
→ znumberp (elt1 (A$0, I$2))

```
#| (COMMENT INPUT F F T) |#
```

AXIOM: assignment2
(BOOLE$3 = BOOLE$2)
∧ ((CAND$3 = elt1 (A$0, I$2)) ∧ ((I$3 = I$2) ∧ (K$3 = K$2)))

AXIOM: assignment3
(BOOLE$4 = BOOLE$3)
∧ ((CAND$4 = CAND$3) ∧ ((I$4 = I$3) ∧ (K$4 = '1)))

```
(PROVE-LEMMA PHASE1-INVRT NIL
 (IMPLIES
  (AND (ZEQP (K$2) '0)
       (AND (NOT (ZGREATERP (I$2) (N$0)))
            (GLOBAL-HYPS)))
  (AND
   (AND
    (NUMBERP (I$4))
    (AND
     (NOT (EQUAL (I$4) '0))
     (AND
      (NUMBERP (K$4))
      (AND
       (NOT (LESSP (I$4) (K$4)))
```

```
       (AND
        (NOT (LESSP (N$0) (I$4)))
        (AND
         (IMPLIES (NOT (NEGATIVEP (CAND$4)))
                  (NUMBERP (CAND$4)))
         (AND
          (NOT (LESSP (CNT (CAND$4) (A$0) '1 (I$4))
                      (K$4)))
          (AND
           (IMPLIES
                  (ZEQP X (CAND$4))
                  (NOT (LESSP (PLUS (I$4) (K$4))
                              (TIMES '2 (CNT X (A$0) '1 (I$4))))))
           (IMPLIES
            (ZNEQP X (CAND$4))
            (NOT
             (LESSP
                   (I$4)
                   (PLUS (K$4)
                         (TIMES '2
                                (CNT X (A$0) '1 (I$4))))))))))))
   (LEX (CONS (ADD1 (PLUS (N$0)
                          (DIFFERENCE (ADD1 (N$0)) (I$4))))
              'NIL)
        (CONS (ADD1 (ADD1 (PLUS (N$0) (N$0))))
              'NIL)))))
```

```
#| (COMMENT INPUT F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F2)
```

AXIOM: logical-if-f2
t

THEOREM: array-bounds-check-for-a
$((\neg\ \text{zeqp}\,(\text{K\$2},\ \text{'0})) \wedge ((\neg\ \text{zgreaterp}\,(\text{I\$2},\ \text{N\$0})) \wedge \text{GLOBAL-HYPS}))$
$\rightarrow\quad ((\text{'0} < \text{I\$2}) \wedge (\text{N\$0} \not< \text{I\$2}))$

```
#| (COMMENT INPUT F F F) |#
```

THEOREM: definedness
((¬ zeqp (K$2, '0)) ∧ ((¬ zgreaterp (I$2, N$0)) ∧ GLOBAL-HYPS))
→ znumberp (CAND$2)

```
#| (COMMENT INPUT F F F) |#
```

THEOREM: definedness1
((¬ zeqp (K$2, '0)) ∧ ((¬ zgreaterp (I$2, N$0)) ∧ GLOBAL-HYPS))
→ znumberp (elt1 (A$0, I$2))

```
#| (COMMENT INPUT F F F) |#

; (FORTRAN-COMMENT LOGICAL-IF-T)
```

AXIOM: logical-if-t
t

THEOREM: input-cond-of-zplus
(zeqp (CAND$2, elt1 (A$0, I$2))
∧ ((¬ zeqp (K$2, '0)) ∧ ((¬ zgreaterp (I$2, N$0)) ∧ GLOBAL-HYPS)))
→ expressible-znumberp (zplus (K$2, '1))

```
#| (COMMENT INPUT F F F T) |#
```

AXIOM: assignment2
(BOOLE$3 = BOOLE$2)
∧ ((CAND$3 = CAND$2) ∧ ((I$3 = I$2) ∧ (K$3 = zplus (K$2, '1))))

```
(PROVE-LEMMA PHASE1-INVRT NIL
 (IMPLIES
  (AND (ZEQP (CAND$2) (ELT1 (A$0) (I$2)))
       (AND (NOT (ZEQP (K$2) '0))
            (AND (NOT (ZGREATERP (I$2) (N$0)))
                 (GLOBAL-HYPS))))
  (AND
   (AND
    (NUMBERP (I$3))
    (AND
     (NOT (EQUAL (I$3) '0))
     (AND
      (NUMBERP (K$3))
```

```
      (AND
       (NOT (LESSP (I$3) (K$3)))
       (AND
        (NOT (LESSP (N$0) (I$3)))
        (AND
         (IMPLIES (NOT (NEGATIVEP (CAND$3)))
                  (NUMBERP (CAND$3)))
         (AND
          (NOT (LESSP (CNT (CAND$3) (A$0) '1 (I$3))
                      (K$3)))
          (AND
           (IMPLIES
                  (ZEQP X (CAND$3))
                  (NOT (LESSP (PLUS (I$3) (K$3))
                              (TIMES '2 (CNT X (A$0) '1 (I$3))))))
           (IMPLIES
            (ZNEQP X (CAND$3))
            (NOT
             (LESSP
                   (I$3)
                   (PLUS (K$3)
                         (TIMES '2
                                (CNT X (A$0) '1 (I$3)))))))))))))
   (LEX (CONS (ADD1 (PLUS (N$0)
                          (DIFFERENCE (ADD1 (N$0)) (I$3))))
              'NIL)
        (CONS (ADD1 (ADD1 (PLUS (N$0) (N$0))))
              'NIL)))))

#| (COMMENT INPUT F F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F3)
```

AXIOM: logical-if-f3
t

THEOREM: input-cond-of-zdifference
$((\neg\ \text{zeqp}\,(\text{CAND\$2},\ \text{elt1}\,(\text{A\$0},\ \text{I\$2})))$
$\wedge\quad((\neg\ \text{zeqp}\,(\text{K\$2},\ \texttt{'0}))\wedge((\neg\ \text{zgreaterp}\,(\text{I\$2},\ \text{N\$0}))\wedge\text{GLOBAL-HYPS})))$
$\rightarrow\quad\text{expressible-znumberp}\,(\text{zdifference}\,(\text{K\$2},\ \texttt{'1}))$

```
#| (COMMENT INPUT F F F F) |#
```

AXIOM: assignment2
(BOOLE$3 = BOOLE$2)
∧ ((CAND$3 = CAND$2)
∧ ((I$3 = I$2) ∧ (K$3 = zdifference (K$2, '1))))

```
(PROVE-LEMMA PHASE1-INVRT NIL
 (IMPLIES
  (AND (NOT (ZEQP (CAND$2) (ELT1 (A$0) (I$2))))
       (AND (NOT (ZEQP (K$2) '0))
            (AND (NOT (ZGREATERP (I$2) (N$0)))
                 (GLOBAL-HYPS))))
  (AND
   (AND
    (NUMBERP (I$3))
    (AND
     (NOT (EQUAL (I$3) '0))
     (AND
      (NUMBERP (K$3))
      (AND
       (NOT (LESSP (I$3) (K$3)))
       (AND
        (NOT (LESSP (N$0) (I$3)))
        (AND
         (IMPLIES (NOT (NEGATIVEP (CAND$3)))
                  (NUMBERP (CAND$3)))
         (AND
          (NOT (LESSP (CNT (CAND$3) (A$0) '1 (I$3))
                      (K$3)))
          (AND
           (IMPLIES
                  (ZEQP X (CAND$3))
                  (NOT (LESSP (PLUS (I$3) (K$3))
                              (TIMES '2 (CNT X (A$0) '1 (I$3))))))
           (IMPLIES
            (ZNEQP X (CAND$3))
            (NOT
             (LESSP
                   (I$3)
                   (PLUS (K$3)
```

```
                        (TIMES '2
                               (CNT X (A$0) '1 (I$3)))))))))))))))
   (LEX (CONS (ADD1 (PLUS (N$0)
                          (DIFFERENCE (ADD1 (N$0)) (I$3))))
              'NIL)
        (CONS (ADD1 (ADD1 (PLUS (N$0) (N$0))))
              'NIL)))))
```

```
#| (COMMENT INPUT F F F F) |#
```

EVENT: Undo back through the event named 'input'.

AXIOM: paths-from-phase1-invrt
$(\text{zeqp}\,(x, \text{CAND\$1}) \rightarrow ((\text{I\$1} + \text{K\$1}) \not< (\text{'2} * \text{cnt}\,(x, \text{A\$0}, \text{'1}, \text{I\$1}))))$
$\wedge \quad (\text{zneqp}\,(x, \text{CAND\$1}) \rightarrow (\text{I\$1} \not< (\text{K\$1} + (\text{'2} * \text{cnt}\,(x, \text{A\$0}, \text{'1}, \text{I\$1})))))$

DEFINITION:
PATH-HYPS
$=$ (GLOBAL-HYPS
$\wedge$ ((I\$1 $\in$ **N**)
$\wedge$ ((I\$1 $\neq$ '0)
$\wedge$ ((K\$1 $\in$ **N**)
$\wedge$ ((I\$1 $\not<$ K\$1)
$\wedge$ ((N\$0 $\not<$ I\$1)
$\wedge$ (((¬ negativep (CAND\$1))
$\rightarrow$ (CAND\$1 $\in$ **N**))
$\wedge$ (cnt (CAND\$1,
A\$0,
'1,
I\$1)
$\not<$ K\$1))))))))

THEOREM: definedness2
PATH-HYPS $\rightarrow$ znumberp (I\$1)

```
#| (COMMENT PHASE1-INVRT) |#
```

THEOREM: input-cond-of-zplus
PATH-HYPS $\rightarrow$ expressible-znumberp (zplus (I\$1, '1))

```
#| (COMMENT PHASE1-INVRT) |#
```

AXIOM: assignment3
(BOOLE$2 = BOOLE$1)
∧ ((CAND$2 = CAND$1) ∧ ((I$2 = zplus (I$1, '1)) ∧ (K$2 = K$1)))

```
; (FORTRAN-COMMENT LOGICAL-IF-T)
```

AXIOM: logical-if-t
t

AXIOM: effects-of-undefiner
(BOOLE$3 = BOOLE$2) ∧ ((CAND$3 = CAND$2) ∧ (K$3 = K$2))

THEOREM: definedness3
(I$2 ∧ (zgreaterp (I$2, N$0) ∧ PATH-HYPS)) → znumberp (K$3)

```
#| (COMMENT PHASE1-INVRT T) |#

; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

AXIOM: assignment4
(BOOLE$4 = '*1*false)
∧ ((CAND$4 = CAND$3) ∧ ((I$4 = I$3) ∧ (K$4 = K$3)))

THEOREM: compound-invrt
(K$1 = 0) → (I$1 ≮ (2 ∗ cnt (*x*, A$0, 1, I$1)))

THEOREM: output
(zeqp (K$3, '0) ∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ PATH-HYPS)))
→ (((BOOLE$4 = '*1*true) ∨ (BOOLE$4 = '*1*false))
∧ **if** BOOLE$4
then znumberp (CAND$4)
∧ ((N$0 ÷ '2) < cnt (CAND$4, A$0, '1, N$0))
else (N$0 ÷ '2) ≮ cnt (*x*, A$0, '1, N$0) **endif**)

```
#| (COMMENT PHASE1-INVRT T T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F4)
```

AXIOM: logical-if-f4
t

AXIOM: assignment4
(BOOLE$4 = '*1*true)
∧ ((CAND$4 = CAND$3) ∧ ((I$4 = I$3) ∧ (K$4 = K$3)))

THEOREM: input-cond-of-zquotient
((¬ zeqp (K$3, '0)) ∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ PATH-HYPS)))
→ ((¬ zeqp ('2, '0)) ∧ expressible-znumberp (zquotient (N$0, '2)))

```
#| (COMMENT PHASE1-INVRT T F) |#

; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

THEOREM: output
(zgreaterp (K$4, zquotient (N$0, '2))
∧ ((¬ zeqp (K$3, '0))
∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ PATH-HYPS))))
→ (((BOOLE$4 = '*1*true) ∨ (BOOLE$4 = '*1*false))
∧ **if** BOOLE$4
then znumberp (CAND$4)
∧ ((N$0 ÷ '2) < cnt (CAND$4, A$0, '1, N$0))
else (N$0 ÷ '2) ≮ cnt (*x*, A$0, '1, N$0) **endif**)

```
#| (COMMENT PHASE1-INVRT T F T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F5)
```

AXIOM: logical-if-f5
t

AXIOM: assignment5
(BOOLE$5 = BOOLE$4)
∧ ((CAND$5 = CAND$4) ∧ ((I$5 = I$4) ∧ (K$5 = '0)))

```
; (FORTRAN-COMMENT LOGICAL-IF-F6)
```

AXIOM: logical-if-f6
t

AXIOM: assignment6
(BOOLE$6 = BOOLE$5)
∧ ((CAND$6 = CAND$5) ∧ ((I$6 = '1) ∧ (K$6 = K$5)))

```
; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

AXIOM: effects-of-undefiner1
$(\text{BOOLE\$7} = \text{BOOLE\$6}) \wedge ((\text{CAND\$7} = \text{CAND\$6}) \wedge (\text{K\$7} = \text{K\$6}))$

AXIOM: assignment7
$(\text{BOOLE\$8} = \texttt{'*1*false})$
$\wedge \quad ((\text{CAND\$8} = \text{CAND\$7}) \wedge ((\text{I\$8} = \text{I\$7}) \wedge (\text{K\$8} = \text{K\$7})))$

THEOREM: output
$(\text{I\$6}$
$\wedge \quad (\text{zgreaterp}\,(\text{I\$6},\ \text{N\$0})$
$\wedge \quad ((\neg\ \text{zgreaterp}\,(\text{K\$4},\ \text{zquotient}\,(\text{N\$0},\ \texttt{'2})))$
$\wedge \quad ((\neg\ \text{zeqp}\,(\text{K\$3},\ \texttt{'0}))$
$\wedge \quad (\text{I\$2} \wedge (\text{zgreaterp}\,(\text{I\$2},\ \text{N\$0}) \wedge \text{PATH-HYPS}))))))$
$\rightarrow \quad (((\text{BOOLE\$8} = \texttt{'*1*true}) \vee (\text{BOOLE\$8} = \texttt{'*1*false}))$
$\wedge \quad$ **if** BOOLE\$8
then znumberp (CAND\$8)
$\wedge \quad ((\text{N\$0} \div \texttt{'2}) < \text{cnt}\,(\text{CAND\$8},\ \text{A\$0},\ \texttt{'1},\ \text{N\$0}))$
else $(\text{N\$0} \div \texttt{'2}) \not< \text{cnt}\,(x,\ \text{A\$0},\ \texttt{'1},\ \text{N\$0})$ **endif**)

```
#| (COMMENT PHASE1-INVRT T F F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F7)
```

AXIOM: logical-if-f7
t

THEOREM: array-bounds-check-for-a1
$((\neg\ \text{zgreaterp}\,(\text{I\$6},\ \text{N\$0}))$
$\wedge \quad ((\neg\ \text{zgreaterp}\,(\text{K\$4},\ \text{zquotient}\,(\text{N\$0},\ \texttt{'2})))$
$\wedge \quad ((\neg\ \text{zeqp}\,(\text{K\$3},\ \texttt{'0}))$
$\wedge \quad (\text{I\$2} \wedge (\text{zgreaterp}\,(\text{I\$2},\ \text{N\$0}) \wedge \text{PATH-HYPS})))))$
$\rightarrow \quad ((\texttt{'0} < \text{I\$6}) \wedge (\text{N\$0} \not< \text{I\$6}))$

```
#| (COMMENT PHASE1-INVRT T F F F F) |#
```

THEOREM: definedness4
((¬ zgreaterp (I$6, N$0))
 ∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
  ∧ ((¬ zeqp (K$3, '0))
   ∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ PATH-HYPS)))))
→ znumberp (CAND$6)

```
#| (COMMENT PHASE1-INVRT T F F F F) |#
```

THEOREM: definedness5
((¬ zgreaterp (I$6, N$0))
 ∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
  ∧ ((¬ zeqp (K$3, '0))
   ∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ PATH-HYPS)))))
→ znumberp (elt1 (A$0, I$6))

```
#| (COMMENT PHASE1-INVRT T F F F F) |#

; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

THEOREM: phase2-invrt
(zneqp (CAND$6, elt1 (A$0, I$6))
 ∧ ((¬ zgreaterp (I$6, N$0))
  ∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
   ∧ ((¬ zeqp (K$3, '0))
    ∧ (I$2 ∧ (zgreaterp (I$2, N$0) ∧ PATH-HYPS))))))
→ (((I$6 ∈ **N**)
  ∧ ((I$6 ≠ '0)
   ∧ ((BOOLE$6 = '*1*true)
    ∧ ((N$0 ≮ I$6)
     ∧ (((¬ negativep (CAND$6))
       → (CAND$6 ∈ **N**))
      ∧ ((zneqp (*x*, CAND$6)
        → (N$0
          ≮ ('2
            ∗ cnt (*x*,
                A$0,
                '1,
                N$0))))
       ∧ ((N$0
          ≮ ('2

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad * \quad \text{cnt}\,(\text{CAND\$6}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{A\$0}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{I\$6}))) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad (\text{K\$6} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{cnt}\,(\text{CAND\$6}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{A\$0}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{I\$6}))))))))) \\
\qquad\qquad\qquad\qquad \wedge \quad \text{lex}\,(\text{cons}\,((1 + \text{N\$0}) - \text{I\$6}, \texttt{'nil}), \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{cons}\,(1 + (\text{N\$0} + ((1 + \text{N\$0}) - \text{I\$1})), \texttt{'nil})))
\end{array}
$$

```
#| (COMMENT PHASE1-INVRT T F F F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F8)
```

AXIOM: logical-if-f8
t

THEOREM: input-cond-of-zplus1
$$
\begin{array}{l}
((\neg\ \text{zneqp}\,(\text{CAND\$6}, \text{elt1}\,(\text{A\$0}, \text{I\$6}))) \\
\ \wedge \quad ((\neg\ \text{zgreaterp}\,(\text{I\$6}, \text{N\$0})) \\
\qquad \wedge \quad ((\neg\ \text{zgreaterp}\,(\text{K\$4}, \text{zquotient}\,(\text{N\$0}, \texttt{'2}))) \\
\qquad\qquad \wedge \quad ((\neg\ \text{zeqp}\,(\text{K\$3}, \texttt{'0})) \\
\qquad\qquad\qquad \wedge \quad (\text{I\$2} \wedge (\text{zgreaterp}\,(\text{I\$2}, \text{N\$0}) \wedge \text{PATH-HYPS})))))) \\
\rightarrow \quad \text{expressible-znumberp}\,(\text{zplus}\,(\text{K\$6}, \texttt{'1}))
\end{array}
$$

```
#| (COMMENT PHASE1-INVRT T F F F F F) |#
```

AXIOM: assignment7
$$
\begin{array}{l}
(\text{BOOLE\$7} = \text{BOOLE\$6}) \\
\wedge \quad ((\text{CAND\$7} = \text{CAND\$6}) \wedge ((\text{I\$7} = \text{I\$6}) \wedge (\text{K\$7} = \text{zplus}\,(\text{K\$6}, \texttt{'1}))))
\end{array}
$$

THEOREM: input-cond-of-zquotient1
$$
\begin{array}{l}
((\neg\ \text{zneqp}\,(\text{CAND\$6}, \text{elt1}\,(\text{A\$0}, \text{I\$6}))) \\
\ \wedge \quad ((\neg\ \text{zgreaterp}\,(\text{I\$6}, \text{N\$0})) \\
\qquad \wedge \quad ((\neg\ \text{zgreaterp}\,(\text{K\$4}, \text{zquotient}\,(\text{N\$0}, \texttt{'2}))) \\
\qquad\qquad \wedge \quad ((\neg\ \text{zeqp}\,(\text{K\$3}, \texttt{'0})) \\
\qquad\qquad\qquad \wedge \quad (\text{I\$2} \wedge (\text{zgreaterp}\,(\text{I\$2}, \text{N\$0}) \wedge \text{PATH-HYPS})))))) \\
\rightarrow \quad ((\neg\ \text{zeqp}\,(\texttt{'2}, \texttt{'0})) \wedge \text{expressible-znumberp}\,(\text{zquotient}\,(\text{N\$0}, \texttt{'2})))
\end{array}
$$

```
#| (COMMENT PHASE1-INVRT T F F F F F) |#

; (FORTRAN-COMMENT LOGICAL-IF-T1)
```

AXIOM: logical-if-t1
t

THEOREM: output
(zgreaterp (K$7, zquotient (N$0, '2))
 ∧ ((¬ zneqp (CAND$6, elt1 (A$0, I$6)))
  ∧ ((¬ zgreaterp (I$6, N$0))
   ∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
    ∧ ((¬ zeqp (K$3, '0))
     ∧ (I$2
      ∧ (zgreaterp (I$2, N$0)
       ∧ PATH-HYPS)))))))
→ (((BOOLE$7 = '*1*true) ∨ (BOOLE$7 = '*1*false))
  ∧ **if** BOOLE$7
   then znumberp (CAND$7)
    ∧ ((N$0 ÷ '2) < cnt (CAND$7, A$0, '1, N$0))
   else (N$0 ÷ '2) ≮ cnt (*x*, A$0, '1, N$0) **endif**)

```
#| (COMMENT PHASE1-INVRT T F F F F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t1'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F9)
```

AXIOM: logical-if-f9
t

THEOREM: phase2-invrt
((¬ zgreaterp (K$7, zquotient (N$0, '2)))
 ∧ ((¬ zneqp (CAND$6, elt1 (A$0, I$6)))
  ∧ ((¬ zgreaterp (I$6, N$0))
   ∧ ((¬ zgreaterp (K$4, zquotient (N$0, '2)))
    ∧ ((¬ zeqp (K$3, '0))
     ∧ (I$2
      ∧ (zgreaterp (I$2, N$0)
       ∧ PATH-HYPS)))))))
→ (((I$7 ∈ **N**)
  ∧ ((I$7 ≠ '0)
   ∧ ((BOOLE$7 = '*1*true)
    ∧ ((N$0 ≮ I$7)

$$
\begin{array}{l}
\wedge \quad (((\neg\ \mathrm{negativep}\,(\textsc{cand}\$7)) \\
\qquad \rightarrow \quad (\textsc{cand}\$7 \in \mathbf{N})) \\
\qquad \wedge \quad ((\mathrm{zneqp}\,(x,\ \textsc{cand}\$7) \\
\qquad\qquad \rightarrow \quad (\textsc{n}\$0 \\
\qquad\qquad\qquad \not< \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad * \quad \mathrm{cnt}\,(x, \\
\qquad\qquad\qquad\qquad\qquad\qquad \textsc{a}\$0, \\
\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \textsc{n}\$0)))) \\
\qquad\qquad \wedge \quad ((\textsc{n}\$0 \\
\qquad\qquad\qquad \not< \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad * \quad \mathrm{cnt}\,(\textsc{cand}\$7, \\
\qquad\qquad\qquad\qquad\qquad\qquad \textsc{a}\$0, \\
\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \textsc{i}\$7))) \\
\qquad\qquad\qquad \wedge \quad (\textsc{k}\$7 \\
\qquad\qquad\qquad\qquad = \quad \mathrm{cnt}\,(\textsc{cand}\$7, \\
\qquad\qquad\qquad\qquad\qquad\qquad \textsc{a}\$0, \\
\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \textsc{i}\$7))))))))) \\
\quad \wedge \quad \mathrm{lex}\,(\mathrm{cons}\,((1 + \textsc{n}\$0) - \textsc{i}\$7,\ \texttt{'nil}), \\
\qquad\qquad \mathrm{cons}\,(1 + (\textsc{n}\$0 + ((1 + \textsc{n}\$0) - \textsc{i}\$1)),\ \texttt{'nil})))
\end{array}
$$

```
#| (COMMENT PHASE1-INVRT T F F F F F F) |#
```

EVENT: Undo back through the event named 'logical-if-t'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F4)
```

AXIOM: logical-if-f4
t

THEOREM: definedness3
$((\neg\ \mathrm{zgreaterp}\,(\textsc{i}\$2,\ \textsc{n}\$0)) \wedge \textsc{path-hyps}) \rightarrow \mathrm{znumberp}\,(\textsc{k}\$2)$

```
#| (COMMENT PHASE1-INVRT F) |#
```

```
; (FORTRAN-COMMENT LOGICAL-IF-T)
```

AXIOM: logical-if-t
t

THEOREM: array-bounds-check-for-a1
$(\mathrm{zeqp}\,(\textsc{k}\$2,\ \texttt{'0}) \wedge ((\neg\ \mathrm{zgreaterp}\,(\textsc{i}\$2,\ \textsc{n}\$0)) \wedge \textsc{path-hyps}))$
$\rightarrow \quad ((\texttt{'0} < \textsc{i}\$2) \wedge (\textsc{n}\$0 \not< \textsc{i}\$2))$

```
#| (COMMENT PHASE1-INVRT F T) |#
```

THEOREM: definedness4
(zeqp (K$2, '0) ∧ ((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS))
→ znumberp (elt1 (A$0, I$2))

```
#| (COMMENT PHASE1-INVRT F T) |#
```

AXIOM: assignment4
(BOOLE$3 = BOOLE$2)
∧ ((CAND$3 = elt1 (A$0, I$2)) ∧ ((I$3 = I$2) ∧ (K$3 = K$2)))

AXIOM: assignment5
(BOOLE$4 = BOOLE$3)
∧ ((CAND$4 = CAND$3) ∧ ((I$4 = I$3) ∧ (K$4 = '1)))

THEOREM: compound-invrt
(K$1 = 0) → (I$1 ≮ (2 ∗ cnt (*x*, A$0, 1, I$1)))

```
(PROVE-LEMMA PHASE1-INVRT1 NIL
 (IMPLIES
  (AND (ZEQP (K$2) '0)
       (AND (NOT (ZGREATERP (I$2) (N$0)))
            (PATH-HYPS)))
  (AND
   (AND
    (NUMBERP (I$4))
    (AND
     (NOT (EQUAL (I$4) '0))
     (AND
      (NUMBERP (K$4))
      (AND
       (NOT (LESSP (I$4) (K$4)))
       (AND
        (NOT (LESSP (N$0) (I$4)))
        (AND
         (IMPLIES (NOT (NEGATIVEP (CAND$4)))
                  (NUMBERP (CAND$4)))
         (AND
          (NOT (LESSP (CNT (CAND$4) (A$0) '1 (I$4))
                      (K$4)))
```

```
          (AND
           (IMPLIES
                  (ZEQP X (CAND$4))
                  (NOT (LESSP (PLUS (I$4) (K$4))
                              (TIMES '2 (CNT X (A$0) '1 (I$4))))))
           (IMPLIES
            (ZNEQP X (CAND$4))
            (NOT
             (LESSP
                   (I$4)
                   (PLUS (K$4)
                         (TIMES '2
                                (CNT X (A$0) '1 (I$4))))))))))
   (LEX (CONS (ADD1 (PLUS (N$0)
                          (DIFFERENCE (ADD1 (N$0)) (I$4))))
              'NIL)
        (CONS (ADD1 (PLUS (N$0)
                          (DIFFERENCE (ADD1 (N$0)) (I$1))))
              'NIL)))))
```

```
#| (COMMENT PHASE1-INVRT F T) |#
```

EVENT: Undo back through the event named 'logical-if-t'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F5)
```

AXIOM: logical-if-f5
t

THEOREM: array-bounds-check-for-a1
$((\neg\ \text{zeqp}\,(\text{K\$2},\ \texttt{'0})) \wedge ((\neg\ \text{zgreaterp}\,(\text{I\$2},\ \text{N\$0})) \wedge \text{PATH-HYPS}))$
$\rightarrow\quad ((\texttt{'0} < \text{I\$2}) \wedge (\text{N\$0} \not< \text{I\$2}))$

```
#| (COMMENT PHASE1-INVRT F F) |#
```

THEOREM: definedness4
$((\neg\ \text{zeqp}\,(\text{K\$2},\ \texttt{'0})) \wedge ((\neg\ \text{zgreaterp}\,(\text{I\$2},\ \text{N\$0})) \wedge \text{PATH-HYPS}))$
$\rightarrow\quad \text{znumberp}\,(\text{CAND\$2})$

```
#| (COMMENT PHASE1-INVRT F F) |#
```

THEOREM: definedness5
$((\neg\ \text{zeqp}\,(\text{K\$2},\ \texttt{'0})) \wedge ((\neg\ \text{zgreaterp}\,(\text{I\$2},\ \text{N\$0})) \wedge \text{PATH-HYPS}))$
$\rightarrow\quad \text{znumberp}\,(\text{elt1}\,(\text{A\$0},\ \text{I\$2}))$

```
#| (COMMENT PHASE1-INVRT F F) |#

; (FORTRAN-COMMENT LOGICAL-IF-T)
```

AXIOM: logical-if-t
t

THEOREM: input-cond-of-zplus1
(zeqp (CAND$2, elt1 (A$0, I$2))
 ∧ ((¬ zeqp (K$2, '0)) ∧ ((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS)))
→ expressible-znumberp (zplus (K$2, '1))

```
#| (COMMENT PHASE1-INVRT F F T) |#
```

AXIOM: assignment4
(BOOLE$3 = BOOLE$2)
∧ ((CAND$3 = CAND$2) ∧ ((I$3 = I$2) ∧ (K$3 = zplus (K$2, '1))))

```
(PROVE-LEMMA PHASE1-INVRT1 NIL
 (IMPLIES
  (AND (ZEQP (CAND$2) (ELT1 (A$0) (I$2)))
       (AND (NOT (ZEQP (K$2) '0))
            (AND (NOT (ZGREATERP (I$2) (N$0)))
                 (PATH-HYPS))))
  (AND
   (AND
    (NUMBERP (I$3))
    (AND
     (NOT (EQUAL (I$3) '0))
     (AND
      (NUMBERP (K$3))
      (AND
       (NOT (LESSP (I$3) (K$3)))
       (AND
        (NOT (LESSP (N$0) (I$3)))
        (AND
         (IMPLIES (NOT (NEGATIVEP (CAND$3)))
                  (NUMBERP (CAND$3)))
         (AND
          (NOT (LESSP (CNT (CAND$3) (A$0) '1 (I$3))
                      (K$3)))
          (AND
```

```
           (IMPLIES
                  (ZEQP X (CAND$3))
                  (NOT (LESSP (PLUS (I$3) (K$3))
                              (TIMES '2 (CNT X (A$0) '1 (I$3))))))
           (IMPLIES
            (ZNEQP X (CAND$3))
            (NOT
             (LESSP
                   (I$3)
                   (PLUS (K$3)
                         (TIMES '2
                                (CNT X (A$0) '1 (I$3)))))))))))))))
   (LEX (CONS (ADD1 (PLUS (N$0)
                          (DIFFERENCE (ADD1 (N$0)) (I$3))))
              'NIL)
        (CONS (ADD1 (PLUS (N$0)
                          (DIFFERENCE (ADD1 (N$0)) (I$1))))
              'NIL)))))
```

```
#| (COMMENT PHASE1-INVRT F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F6)
```

AXIOM: logical-if-f6
t

THEOREM: input-cond-of-zdifference1
((¬ zeqp (CAND$2, elt1 (A$0, I$2)))
 ∧ ((¬ zeqp (K$2, '0)) ∧ ((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS)))
→ expressible-znumberp (zdifference (K$2, '1))

```
#| (COMMENT PHASE1-INVRT F F F) |#
```

AXIOM: assignment4
(BOOLE$3 = BOOLE$2)
∧ ((CAND$3 = CAND$2)
   ∧ ((I$3 = I$2) ∧ (K$3 = zdifference (K$2, '1)))))

```
(PROVE-LEMMA PHASE1-INVRT1 NIL
```

```
 (IMPLIES
  (AND (NOT (ZEQP (CAND$2) (ELT1 (A$0) (I$2))))
       (AND (NOT (ZEQP (K$2) '0))
            (AND (NOT (ZGREATERP (I$2) (N$0)))
                 (PATH-HYPS))))
  (AND
   (AND
    (NUMBERP (I$3))
    (AND
     (NOT (EQUAL (I$3) '0))
     (AND
      (NUMBERP (K$3))
      (AND
       (NOT (LESSP (I$3) (K$3)))
       (AND
        (NOT (LESSP (N$0) (I$3)))
        (AND
         (IMPLIES (NOT (NEGATIVEP (CAND$3)))
                  (NUMBERP (CAND$3)))
         (AND
          (NOT (LESSP (CNT (CAND$3) (A$0) '1 (I$3))
                      (K$3)))
          (AND
           (IMPLIES
                  (ZEQP X (CAND$3))
                  (NOT (LESSP (PLUS (I$3) (K$3))
                              (TIMES '2 (CNT X (A$0) '1 (I$3))))))
           (IMPLIES
            (ZNEQP X (CAND$3))
            (NOT
             (LESSP
                   (I$3)
                   (PLUS (K$3)
                         (TIMES '2
                                (CNT X (A$0) '1 (I$3))))))))))))))
   (LEX (CONS (ADD1 (PLUS (N$0)
                          (DIFFERENCE (ADD1 (N$0)) (I$3))))
              'NIL)
        (CONS (ADD1 (PLUS (N$0)
                          (DIFFERENCE (ADD1 (N$0)) (I$1))))
              'NIL)))))

#| (COMMENT PHASE1-INVRT F F F) |#
```

EVENT: Undo back through the event named 'paths-from-phase1-invrt'.

AXIOM: paths-from-phase2-invrt
(BOOLE$1 = '*1*true)
∧ ((zneqp (x, CAND$1) → (N$0 ≮ ('2 ∗ cnt (x, A$0, '1, N$0))))
∧ (K$1 = cnt (CAND$1, A$0, '1, I$1)))

DEFINITION:
PATH-HYPS
= (GLOBAL-HYPS
∧ ((I$1 ∈ **N**)
∧ ((I$1 ≠ '0)
∧ ((N$0 ≮ I$1)
∧ (((¬ negativep (CAND$1))
→ (CAND$1 ∈ **N**))
∧ (N$0
≮ ('2
∗ cnt (CAND$1,
A$0,
'1,
I$1)))))))))

THEOREM: definedness2
PATH-HYPS → znumberp (I$1)

```
#| (COMMENT PHASE2-INVRT) |#
```

THEOREM: input-cond-of-zplus
PATH-HYPS → expressible-znumberp (zplus (I$1, '1))

```
#| (COMMENT PHASE2-INVRT) |#
```

AXIOM: assignment3
(BOOLE$2 = BOOLE$1)
∧ ((CAND$2 = CAND$1) ∧ ((I$2 = zplus (I$1, '1)) ∧ (K$2 = K$1)))

```
; (FORTRAN-COMMENT LOGICAL-IF-T)
```

AXIOM: logical-if-t
t

AXIOM: effects-of-undefiner
(BOOLE$3 = BOOLE$2) ∧ ((CAND$3 = CAND$2) ∧ (K$3 = K$2))

AXIOM: assignment4
(BOOLE$4 = '*1*false)
∧ ((CAND$4 = CAND$3) ∧ ((I$4 = I$3) ∧ (K$4 = K$3)))

THEOREM: zeqp-is-a-congruence-relation-wrt-cnt-arg-1
(znormalize (x) = znormalize (y)) → (cnt (x, a, i, j) ≮ cnt (y, a, i, j))

THEOREM: phase-2-hint
(N$0 ≮ (cnt (CAND$1, A$0, 1, N$0) + cnt (CAND$1, A$0, 1, N$0)))
→ (N$0 ≮ (cnt (x, A$0, 1, N$0) + cnt (x, A$0, 1, N$0)))

THEOREM: output
(I$2 ∧ (zgreaterp (I$2, N$0) ∧ PATH-HYPS))
→ (((BOOLE$4 = '*1*true) ∨ (BOOLE$4 = '*1*false))
∧ **if** BOOLE$4
then znumberp (CAND$4)
∧ ((N$0 ÷ '2) < cnt (CAND$4, A$0, '1, N$0))
else (N$0 ÷ '2) ≮ cnt (x, A$0, '1, N$0) **endif**)

```
#| (COMMENT PHASE2-INVRT T) |#
```

EVENT: Undo back through the event named 'logical-if-t'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F4)
```

AXIOM: logical-if-f4
t

THEOREM: array-bounds-check-for-a1
((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS)
→ ('0 < I$2) ∧ (N$0 ≮ I$2))

```
#| (COMMENT PHASE2-INVRT F) |#
```

THEOREM: definedness3
((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS) → znumberp (CAND$2)

```
#| (COMMENT PHASE2-INVRT F) |#
```

THEOREM: definedness4
((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS) → znumberp (elt1 (A$0, I$2))

```
#| (COMMENT PHASE2-INVRT F) |#

; (FORTRAN-COMMENT LOGICAL-IF-T)
```

AXIOM: logical-if-t
t

THEOREM: phase2-invrt1

$$
\begin{array}{l}
(\text{zneqp}\,(\text{CAND\$2},\ \text{elt1}\,(\text{A\$0},\ \text{I\$2})) \wedge ((\neg\ \text{zgreaterp}\,(\text{I\$2},\ \text{N\$0})) \wedge \text{PATH-HYPS})) \\
\rightarrow\quad (((\text{I\$2} \in \mathbf{N}) \\
\qquad \wedge\quad ((\text{I\$2} \neq \texttt{'0}) \\
\qquad\qquad \wedge\quad ((\text{BOOLE\$2} = \texttt{'*1*true}) \\
\qquad\qquad\qquad \wedge\quad ((\text{N\$0} \not< \text{I\$2}) \\
\qquad\qquad\qquad\qquad \wedge\quad (((\neg\ \text{negativep}\,(\text{CAND\$2})) \\
\qquad\qquad\qquad\qquad\qquad \rightarrow\quad (\text{CAND\$2} \in \mathbf{N})) \\
\qquad\qquad\qquad\qquad\qquad \wedge\quad ((\text{zneqp}\,(x,\ \text{CAND\$2}) \\
\qquad\qquad\qquad\qquad\qquad\qquad \rightarrow\quad (\text{N\$0} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \not< \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad *\quad \text{cnt}\,(x, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{A\$0}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{N\$0})))) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge\quad ((\text{N\$0} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \not< \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad *\quad \text{cnt}\,(\text{CAND\$2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{A\$0}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{I\$2}))) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge\quad (\text{K\$2} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad =\quad \text{cnt}\,(\text{CAND\$2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{A\$0}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{I\$2}))))))))) \\
\qquad \wedge\quad \text{lex}\,(\text{cons}\,((1 + \text{N\$0}) - \text{I\$2},\ \texttt{'nil}), \\
\qquad\qquad\quad \text{cons}\,((1 + \text{N\$0}) - \text{I\$1},\ \texttt{'nil})))
\end{array}
$$

```
#| (COMMENT PHASE2-INVRT F T) |#
```

EVENT: Undo back through the event named 'logical-if-t'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F5)
```

AXIOM: logical-if-f5
t

THEOREM: definedness5
((¬ zneqp (CAND$2, elt1 (A$0, I$2)))
 ∧ ((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS))
→ znumberp (K$2)

```
#| (COMMENT PHASE2-INVRT F F) |#
```

THEOREM: input-cond-of-zplus1
((¬ zneqp (CAND$2, elt1 (A$0, I$2)))
 ∧ ((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS))
→ expressible-znumberp (zplus (K$2, '1))

```
#| (COMMENT PHASE2-INVRT F F) |#
```

AXIOM: assignment4
(BOOLE$3 = BOOLE$2)
∧ ((CAND$3 = CAND$2) ∧ ((I$3 = I$2) ∧ (K$3 = zplus (K$2, '1))))

THEOREM: cnt-grows
((I$1 < n) ∧ zeqp (x, elt1 (a, 1 + I$1)))
→ (cnt (x, a, 1, I$1) < cnt ($x$, a, 1, n))

THEOREM: input-cond-of-zquotient
((¬ zneqp (CAND$2, elt1 (A$0, I$2)))
 ∧ ((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS))
→ ((¬ zeqp ('2, '0)) ∧ expressible-znumberp (zquotient (N$0, '2)))

```
#| (COMMENT PHASE2-INVRT F F) |#
```

```
; (FORTRAN-COMMENT LOGICAL-IF-T)
```

AXIOM: logical-if-t
t

THEOREM: output
(zgreaterp (K$3, zquotient (N$0, '2))
 ∧ ((¬ zneqp (CAND$2, elt1 (A$0, I$2)))
  ∧ ((¬ zgreaterp (I$2, N$0)) ∧ PATH-HYPS)))
→ (((BOOLE$3 = '*1*true) ∨ (BOOLE$3 = '*1*false))
  ∧ **if** BOOLE$3
    then znumberp (CAND$3)
      ∧ ((N$0 ÷ '2) < cnt (CAND$3, A$0, '1, N$0))
    else (N$0 ÷ '2) ≮ cnt ($x$, A$0, '1, N$0) **endif**)

```
#| (COMMENT PHASE2-INVRT F F T) |#
```

EVENT: Undo back through the event named 'logical-if-t'.

```
; (FORTRAN-COMMENT LOGICAL-IF-F6)
```

AXIOM: logical-if-f6
t

THEOREM: phase2-invrt1

$$
\begin{array}{l}
((\neg\ \mathrm{zgreaterp}\,(\mathrm{K}\$3,\ \mathrm{zquotient}\,(\mathrm{N}\$0,\ \texttt{'2}))) \\
\ \wedge\quad ((\neg\ \mathrm{zneqp}\,(\mathrm{CAND}\$2,\ \mathrm{elt1}\,(\mathrm{A}\$0,\ \mathrm{I}\$2))) \\
\qquad \wedge\quad ((\neg\ \mathrm{zgreaterp}\,(\mathrm{I}\$2,\ \mathrm{N}\$0))\ \wedge\ \text{PATH-HYPS}))) \\
\rightarrow\quad (((\mathrm{I}\$3 \in \mathbf{N}) \\
\qquad \wedge\quad ((\mathrm{I}\$3 \neq \texttt{'0}) \\
\qquad\qquad \wedge\quad ((\mathrm{BOOLE}\$3 = \texttt{'*1*true}) \\
\qquad\qquad\qquad \wedge\quad ((\mathrm{N}\$0 \not< \mathrm{I}\$3) \\
\qquad\qquad\qquad\qquad \wedge\quad (((\neg\ \mathrm{negativep}\,(\mathrm{CAND}\$3)) \\
\qquad\qquad\qquad\qquad\qquad \rightarrow\quad (\mathrm{CAND}\$3 \in \mathbf{N})) \\
\qquad\qquad\qquad\qquad\qquad \wedge\quad ((\mathrm{zneqp}\,(x,\ \mathrm{CAND}\$3) \\
\qquad\qquad\qquad\qquad\qquad\qquad \rightarrow\quad (\mathrm{N}\$0 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \not< \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad * \quad \mathrm{cnt}\,(x, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{A}\$0, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{N}\$0)))) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge\quad ((\mathrm{N}\$0 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \not< \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad * \quad \mathrm{cnt}\,(\mathrm{CAND}\$3, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{A}\$0, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{I}\$3))) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge\quad (\mathrm{K}\$3 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \mathrm{cnt}\,(\mathrm{CAND}\$3, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{A}\$0, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathrm{I}\$3))))))))) \\
\qquad \wedge\quad \mathrm{lex}\,(\mathrm{cons}\,((1 + \mathrm{N}\$0) - \mathrm{I}\$3,\ \texttt{'nil}), \\
\qquad\qquad\quad \mathrm{cons}\,((1 + \mathrm{N}\$0) - \mathrm{I}\$1,\ \texttt{'nil})))
\end{array}
$$

```
#| (COMMENT PHASE2-INVRT F F F) |#
```

EVENT: Undo back through the event named 'paths-from-phase2-invrt'.

EVENT: Undo back through the event named 'fortran'.

```
#|

The correctness of the program depends upon the following events:

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Definition.
        (CNT X A I N)
           =
        (IF (OR (ZEROP N) (LESSP N I))
            0
            (IF (ZEQP X (ELT1 A N))
                (ADD1 (CNT X A I (SUB1 N)))
                (CNT X A I (SUB1 N))))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        (FORTRAN-COMMENT USEFUL-LEMMAS)
@END(VERBATIM)
@END(GROUP)
```

```
Specification for routine MJRTY

The input assertion:
        (AND (IMPLIES (AND (LESSP 0 J)
                           (NOT (LESSP (N STATE) J))
                           (NOT (NEGATIVEP (ELT1 A J))))
                      (NUMBERP (ELT1 A J)))
             (NUMBERP (N STATE))
             (NOT (EQUAL (N STATE) 0))
             (LESSP (ADD1 (N STATE))
                    (LEAST-INEXPRESSIBLE-POSITIVE-INTEGER)))


The output assertion:
        (AND (OR (EQUAL (BOOLE NEWSTATE) (TRUE))
                 (EQUAL (BOOLE NEWSTATE) (FALSE)))
             (IF (BOOLE NEWSTATE)
                 (AND (ZNUMBERP (CAND NEWSTATE))
                      (LESSP (QUOTIENT (N STATE) 2)
                             (CNT (CAND NEWSTATE)
                                  (A STATE)
                                  1
                                  (N STATE))))
                 (NOT (LESSP (QUOTIENT (N STATE) 2)
                             (CNT X (A STATE) 1 (N STATE))))))
```

```
      END
      SUBROUTINE MJRTY(A, N, BOOLE, CAND)
      INTEGER N
      INTEGER A
      LOGICAL BOOLE
      INTEGER CAND
      INTEGER I
      INTEGER K
      DIMENSION A(N)
      K = 0
C     THE FOLLOWING DO IMPLEMENTS THE PAIRING PHASE. CAND IS THE CURRENT
C     LY LEADING CANDIDATE AND K IS THE NUMBER OF UNPAIRED VOTES FOR CAN
C     D.
      DO 100 I = 1, N
C     DOJUNK PHASE1-HINT
      IF ((K .EQ. 0)) GOTO 50
      IF ((CAND .EQ. A(I))) GOTO 75
      K = (K - 1)
      GOTO 100
50    CAND = A(I)
      K = 1
C     XXX PHASE1-INVRT-F-T
      GOTO 100
75    K = (K + 1)
100   CONTINUE
      IF ((K .EQ. 0)) GOTO 300
      BOOLE = .TRUE.
      IF ((K .GT. (N / 2))) RETURN
C     WE NOW ENTER THE COUNTING PHASE. BOOLE IS SET TO TRUE IN ANTICIPAT
C     ION OF FINDING CAND IN THE MAJORITY. K IS USED AS THE RUNNING TALL
C     Y FOR CAND. WE EXIT AS SOON AS K EXCEEDS N/2.
      K = 0
      DO 200 I = 1, N
C     DOJUNK PHASE2
      IF ((CAND .NE. A(I))) GOTO 200
      K = (K + 1)
C     XXX OUTPUT-HINT
      IF ((K .GT. (N / 2))) RETURN
200   CONTINUE
300   BOOLE = .FALSE.
C     XXX HINT-FOR-PHASE1-INVRT-T-T
C     XXX HINT-FOR-PHASE2-INVRT-T
      RETURN
```

```
      END


The XXX at USEFUL-LEMMAS.
@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  PLUS-1 (rewrite):
        (EQUAL (PLUS 1 X) (ADD1 X))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  DIFFERENCE-0 (rewrite):
        (IMPLIES (NOT (LESSP X Y))
                 (EQUAL (DIFFERENCE Y X) 0))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  DIFFERENCE-1 (rewrite):
        (EQUAL (DIFFERENCE X 1) (SUB1 X))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  LESSP-X-1 (rewrite):
        (EQUAL (LESSP X 1) (ZEROP X))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  LESSP-REMAINDER2 (rewrite and generalize):
        (EQUAL (LESSP (REMAINDER X Y) Y)
               (NOT (ZEROP Y)))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  REMAINDER-QUOTIENT-ELIM (elimination):
```

```
        (IMPLIES (AND (NOT (ZEROP Y)) (NUMBERP X))
                 (EQUAL (PLUS (REMAINDER X Y)
                              (TIMES Y (QUOTIENT X Y)))
                        X))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  QUOTIENT-BY-2-BOUND (rewrite):
        (NOT (LESSP X (QUOTIENT X 2)))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  LESSP-QUOTIENT-REWRITE (rewrite):
        (EQUAL (LESSP (QUOTIENT N 2) M)
               (LESSP N (PLUS M M)))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  ZNORMALIZE-ZERO (rewrite):
        (IMPLIES (NUMBERP X)
                 (EQUAL (EQUAL (ZNORMALIZE X) 0)
                        (EQUAL X 0)))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Enable ZNORMALIZE.
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  CNT-BOUND (rewrite):
        (NOT (LESSP N (CNT X A 1 N)))
@END(VERBATIM)
@END(GROUP)
```

```
Hints for routine MJRTY



The input clock:
        (LIST (ADD1 (ADD1 (PLUS (N (START)) (N (START))))))

The DO junk named PHASE1-HINT:
        ((BUMP PHASE1-INVRT))

The invariant and clock named PHASE1-INVRT.

        (AND
         (NUMBERP (I STATE))
         (NOT (EQUAL (I STATE) 0))
         (NUMBERP (K STATE))
         (NOT (LESSP (I STATE) (K STATE)))
         (NOT (LESSP (N (START)) (I STATE)))
         (IMPLIES (NOT (NEGATIVEP (CAND STATE)))
                  (NUMBERP (CAND STATE)))
         (NOT (LESSP (CNT (CAND STATE)
                          (A STATE)
                          1
                          (I STATE))
                     (K STATE)))
         (IMPLIES
                (ZEQP X (CAND STATE))
                (NOT (LESSP (PLUS (I STATE) (K STATE))
                            (TIMES 2
                                   (CNT X (A STATE) 1 (I STATE))))))
         (IMPLIES
          (ZNEQP X (CAND STATE))
          (NOT
             (LESSP (I STATE)
                    (PLUS (K STATE)
                          (TIMES 2
                                 (CNT X (A STATE) 1 (I STATE))))))))

        (LIST (ADD1 (PLUS (N (START))
                          (DIFFERENCE (ADD1 (N (START)))
                                      (I STATE)))))

The XXX named PHASE1-INVRT-F-T:
```

```
        Path encryption:
        ((PHASE1-INVRT F T))
@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  COMPOUND-INVRT (rewrite):
        (IMPLIES (EQUAL (K$1) 0)
                 (NOT (LESSP (I$1)
                             (TIMES 2 (CNT X (A$0) 1 (I$1))))))
        Hints:  Consider:
                 PATHS-FROM-PHASE1-INVRT
                Enable  PATHS-FROM-PHASE1-INVRT
@END(VERBATIM)
@END(GROUP)



The DO junk named PHASE2:
        ((BUMP PHASE2-INVRT))

The invariant and clock named PHASE2-INVRT.

        (AND
         (NUMBERP (I STATE))
         (NOT (EQUAL (I STATE) 0))
         (EQUAL (BOOLE STATE) (TRUE))
         (NOT (LESSP (N (START)) (I STATE)))
         (IMPLIES (NOT (NEGATIVEP (CAND STATE)))
                  (NUMBERP (CAND STATE)))
         (IMPLIES
              (ZNEQP X (CAND STATE))
              (NOT (LESSP (N (START))
                          (TIMES 2
                                 (CNT X (A STATE) 1 (N (START)))))))
         (NOT (LESSP (N (START))
                     (TIMES 2
                            (CNT (CAND STATE)
                                 (A STATE)
                                 1
                                 (I STATE)))))
         (EQUAL (K STATE)
                (CNT (CAND STATE)
                     (A STATE)
                     1
                     (I STATE))))
```

```
        (LIST (DIFFERENCE (ADD1 (N (START)))
                          (I STATE)))

The XXX named OUTPUT-HINT:
        Path encryption:
        ((PHASE2-INVRT F F))
@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  CNT-GROWS (rewrite):
        (IMPLIES (AND (LESSP (I$1) N)
                      (ZEQP X (ELT1 A (ADD1 (I$1)))))
                 (LESSP (CNT X A 1 (I$1))
                        (CNT X A 1 N)))
        Hint:  Induct as for (PLUS N I).
@END(VERBATIM)
@END(GROUP)



The XXX named HINT-FOR-PHASE1-INVRT-T-T:
        Path encryption:
        ((PHASE1-INVRT T T))
@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  COMPOUND-INVRT (rewrite):
        (IMPLIES (EQUAL (K$1) 0)
                 (NOT (LESSP (I$1)
                             (TIMES 2 (CNT X (A$0) 1 (I$1))))))
        Hints:  Consider:
                 PATHS-FROM-PHASE1-INVRT
                Enable PATHS-FROM-PHASE1-INVRT
@END(VERBATIM)
@END(GROUP)



The XXX named HINT-FOR-PHASE2-INVRT-T:
        Path encryption:
        ((PHASE2-INVRT T))
@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  ZEQP-IS-A-CONGRUENCE-RELATION-WRT-CNT-ARG-1 (rewrite):
        (IMPLIES (EQUAL (ZNORMALIZE X) (ZNORMALIZE Y))
```

```
                (NOT (LESSP (CNT X A I J) (CNT Y A I J))))
@END(VERBATIM)
@END(GROUP)

@BEGIN(GROUP)
@BEGIN(VERBATIM)
        Theorem.  PHASE-2-HINT (rewrite):
        (IMPLIES (NOT (LESSP (N$0)
                             (PLUS (CNT (CAND$1) (A$0) 1 (N$0))
                                   (CNT (CAND$1) (A$0) 1 (N$0)))))
                 (NOT (LESSP (N$0)
                             (PLUS (CNT X (A$0) 1 (N$0))
                                   (CNT X (A$0) 1 (N$0))))))
        Hints:  Consider:
                 PATHS-FROM-PHASE2-INVRT
                Enable PATHS-FROM-PHASE2-INVRT
@END(VERBATIM)
@END(GROUP)

|#
```

Index