```
#|

 Copyright (C) 1994 by Yuan Yu.  All Rights Reserved.

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

 NO WARRANTY

 Yuan Yu PROVIDES 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 Yuan Yu 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.

|#
```

EVENT: Start with the library "mc20-2" using the compiled version.

```
;          Proof of the Corectness of a Majority Voting Program
;
#|
The following C function MJRTY determines if there is a candidate who
has received a majority of votes cast in an election.

/* a majority voting algorithm invented by Boyer and Moore */
#define YES 1
#define NO  0

struct winner {
  int x;
  int y;
};

struct winner mjrty (int a[], int n)
```

```
{
  int cand, i, k;
  struct winner temp;

  k = 0;
  for (i = 0; i < n; i++)
    if (k == 0) {
      cand = a[i];
      k = 1;
    }
  else {
    if (cand == a[i])
      k++;
    else
      k--;
  };
  temp.x = cand;
  if (k == 0) {
    temp.y = NO;
    return temp;
  };
  if (k > n/2) {
    temp.y = YES;
    return temp;
  };
  k = 0;
  for (i = 0; i < n; i++)
    if (a[i] == cand)
      k++;
  if (k > n/2)
    temp.y = YES;
  else temp.y = NO;
  return temp;
}
```

Here is the MC68020 assembly code of the above C program. The code is generated by "gcc -O".

```
0x2310 <mjrty>:         linkw a6,#0
0x2314 <mjrty+4>:       moveml d2-d5,sp@-
0x2318 <mjrty+8>:       moveal a6@(8),a0
0x231c <mjrty+12>:      movel a6@(12),d2
0x2320 <mjrty+16>:      clrl d1
0x2322 <mjrty+18>:      clrl d0
```

```
0x2324 <mjrty+20>:      cmpl d0,d2
0x2326 <mjrty+22>:      ble 0x2346 <mjrty+54>
0x2328 <mjrty+24>:      tstl d1
0x232a <mjrty+26>:      bne 0x2334 <mjrty+36>
0x232c <mjrty+28>:      movel 0(a0)[d0.l*4],d3
0x2330 <mjrty+32>:      movel #1,d1
0x2332 <mjrty+34>:      bra 0x2340 <mjrty+48>
0x2334 <mjrty+36>:      cmpl 0(a0)[d0.l*4],d3
0x2338 <mjrty+40>:      bne 0x233e <mjrty+46>
0x233a <mjrty+42>:      addql #1,d1
0x233c <mjrty+44>:      bra 0x2340 <mjrty+48>
0x233e <mjrty+46>:      subl #1,d1
0x2340 <mjrty+48>:      addql #1,d0
0x2342 <mjrty+50>:      cmpl d0,d2
0x2344 <mjrty+52>:      bgt 0x2328 <mjrty+24>
0x2346 <mjrty+54>:      movel d3,d4
0x2348 <mjrty+56>:      tstl d1
0x234a <mjrty+58>:      beq 0x2382 <mjrty+114>
0x234c <mjrty+60>:      movel d2,d0
0x234e <mjrty+62>:      bge 0x2352 <mjrty+66>
0x2350 <mjrty+64>:      addql #1,d0
0x2352 <mjrty+66>:      asrl #1,d0
0x2354 <mjrty+68>:      cmpl d1,d0
0x2356 <mjrty+70>:      bge 0x235c <mjrty+76>
0x2358 <mjrty+72>:      movel #1,d5
0x235a <mjrty+74>:      bra 0x2384 <mjrty+116>
0x235c <mjrty+76>:      clrl d1
0x235e <mjrty+78>:      clrl d0
0x2360 <mjrty+80>:      cmpl d0,d2
0x2362 <mjrty+82>:      ble 0x2372 <mjrty+98>
0x2364 <mjrty+84>:      cmpl 0(a0)[d0.l*4],d3
0x2368 <mjrty+88>:      bne 0x236c <mjrty+92>
0x236a <mjrty+90>:      addql #1,d1
0x236c <mjrty+92>:      addql #1,d0
0x236e <mjrty+94>:      cmpl d0,d2
0x2370 <mjrty+96>:      bgt 0x2364 <mjrty+84>
0x2372 <mjrty+98>:      movel d2,d0
0x2374 <mjrty+100>:     bge 0x2378 <mjrty+104>
0x2376 <mjrty+102>:     addql #1,d0
0x2378 <mjrty+104>:     asrl #1,d0
0x237a <mjrty+106>:     cmpl d1,d0
0x237c <mjrty+108>:     bge 0x2382 <mjrty+114>
0x237e <mjrty+110>:     movel #1,d5
0x2380 <mjrty+112>:     bra 0x2384 <mjrty+116>
```

```
0x2382 <mjrty+114>:     clrl d5
0x2384 <mjrty+116>:     movel d4,d0
0x2386 <mjrty+118>:     movel d5,d1
0x2388 <mjrty+120>:     moveml a6@(-16),d2-d5
0x238e <mjrty+126>:     unlk a6
0x2390 <mjrty+128>:     rts

The machine code of the above program is:


<mjrty>:      0x4e56  0x0000  0x48e7  0x3c00  0x206e  0x0008  0x242e  0x000c
<mjrty+16>:   0x4281  0x4280  0xb480  0x6f1e  0x4a81  0x6608  0x2630  0x0c00
<mjrty+32>:   0x7201  0x600c  0xb6b0  0x0c00  0x6604  0x5281  0x6002  0x5381
<mjrty+48>:   0x5280  0xb480  0x6ee2  0x2803  0x4a81  0x6736  0x2002  0x6c02
<mjrty+64>:   0x5280  0xe280  0xb081  0x6c04  0x7a01  0x6028  0x4281  0x4280
<mjrty+80>:   0xb480  0x6f0e  0xb6b0  0x0c00  0x6602  0x5281  0x5280  0xb480
<mjrty+96>:   0x6ef2  0x2002  0x6c02  0x5280  0xe280  0xb081  0x6c04  0x7a01
<mjrty+112>:  0x6002  0x4285  0x2004  0x2205  0x4cee  0x003c  0xfff0  0x4e5e
<mjrty+128>:  0x4e75

In the Nqthm logic, it is like:

'(78     86      0       0       72      231     60      0
  32     110     0       8       36      46      0       12
  66     129     66      128     180     128     111     30
  74     129     102     8       38      48      12      0
  114    1       96      12      182     176     12      0
  102    4       82      129     96      2       83      129
  82     128     180     128     110     226     40      3
  74     129     103     54      32      2       108     2
  82     128     226     128     176     129     108     4
  122    1       96      40      66      129     66      128
  180    128     111     14      182     176     12      0
  102    2       82      129     82      128     180     128
  110    242     32      2       108     2       82      128
  226    128     176     129     108     4       122     1
  96     2       66      133     32      4       34      5
  76     238     0       60      255     240     78      94
  78     117)
|#

; in the logic, the above program is defined by (mjrty-code).
```

DEFINITION:

MJRTY-CODE
= '(78 86 0 0 72 231 60 0 32 110 0 8 36 46 0 12 66 129
66 128 180 128 111 30 74 129 102 8 38 48 12 0 114 1
96 12 182 176 12 0 102 4 82 129 96 2 83 129 82 128
180 128 110 226 40 3 74 129 103 54 32 2 108 2 82 128
226 128 176 129 108 4 122 1 96 40 66 129 66 128 180
128 111 14 182 176 12 0 102 2 82 129 82 128 180 128
110 242 32 2 108 2 82 128 226 128 176 129 108 4 122
1 96 2 66 133 32 4 34 5 76 238 0 60 255 240 78 94 78
117)

```
; mjrty-cand is a function in the logic to simulate the candidate
; findhe above code.
```

DEFINITION:
mjrty-cand $(n,\ lst,\ cand,\ i,\ k)$
$=$ **if** $i < n$
then if $k \simeq 0$ **then** mjrty-cand $(n,\ lst,$ get-nth $(i,\ lst),\ 1 + i,\ 1)$
elseif $cand =$ get-nth $(i,\ lst)$
then mjrty-cand $(n,\ lst,\ cand,\ 1 + i,\ 1 + k)$
else mjrty-cand $(n,\ lst,\ cand,\ 1 + i,\ k - 1)$ **endif**
else $cand$ **endif**

DEFINITION:
mjrty-k $(n,\ lst,\ cand,\ i,\ k)$
$=$ **if** $i < n$
then if $k \simeq 0$ **then** mjrty-k $(n,\ lst,$ get-nth $(i,\ lst),\ 1 + i,\ 1)$
elseif $cand =$ get-nth $(i,\ lst)$
then mjrty-k $(n,\ lst,\ cand,\ 1 + i,\ 1 + k)$
else mjrty-k $(n,\ lst,\ cand,\ 1 + i,\ k - 1)$ **endif**
else k **endif**

```
; cand-cnt is a function in the logic to simulate the process of
; counting the number of votes for the given candidate.
```

DEFINITION:
cand-cnt $(n,\ lst,\ cand,\ i,\ k)$
$=$ **if** $i < n$
then if $cand =$ get-nth $(i,\ lst)$ **then** cand-cnt $(n,\ lst,\ cand,\ 1 + i,\ 1 + k)$
else cand-cnt $(n,\ lst,\ cand,\ 1 + i,\ k)$ **endif**
else k **endif**

```
; mjrty-p determines if the given candidate cand has received a majority
; voting.
```

DEFINITION:
mjrty-p $(n,\ lst,\ cand,\ i,\ k)$
$=$ **if** mjrty-k $(n,\ lst,\ cand,\ i,\ k) \simeq 0$ **then f**
elseif $(n \div 2) <$ mjrty-k $(n,\ lst,\ cand,\ i,\ k)$ **then t**
else $(n \div 2)$
$<$ cand-cnt $(n,\ lst,$ mjrty-cand $(n,\ lst,\ cand,\ i,\ k),\ i,\ k)$ **endif**

```
; the computation time.
```

DEFINITION:
mjrty-cand-t $(a,\ n,\ lst,\ cand,\ i,\ k)$
$=$ **if** $i < n$
then if $k \simeq 0$
then let *cand1* **be** get-nth $(i,\ lst)$
in
splus (8, mjrty-cand-t $(a,\ n,\ lst,\ cand1,\ 1+i,\ 1)$) **endlet**
elseif $cand =$ get-nth $(i,\ lst)$
then splus (9, mjrty-cand-t $(a,\ n,\ lst,\ cand,\ 1+i,\ 1+k)$)
else splus (8, mjrty-cand-t $(a,\ n,\ lst,\ cand,\ 1+i,\ k-1)$) **endif**
elseif $cand =$ get-nth $(0,\ lst)$ **then** 18
else 17 **endif**

DEFINITION:
mjrty-sn-t $(a,\ n,\ lst,\ cand,\ i,\ k)$
$=$ **if** $i < n$
then if $k \simeq 0$
then let *cand1* **be** get-nth $(i,\ lst)$
in
splus (8, mjrty-sn-t $(a,\ n,\ lst,\ cand1,\ 1+i,\ 1)$) **endlet**
elseif $cand =$ get-nth $(i,\ lst)$
then splus (9, mjrty-sn-t $(a,\ n,\ lst,\ cand,\ 1+i,\ 1+k)$)
else splus (8, mjrty-sn-t $(a,\ n,\ lst,\ cand,\ 1+i,\ k-1)$) **endif**
elseif $k \simeq 0$ **then** 11
else 17 **endif**

DEFINITION:
cand-cnt-t $(a,\ n,\ lst,\ cand,\ i,\ k)$
$=$ **if** $i < n$
then if $cand =$ get-nth $(i,\ lst)$
then splus (6, cand-cnt-t $(a,\ n,\ lst,\ cand,\ 1+i,\ 1+k)$)
else splus (5, cand-cnt-t $(a,\ n,\ lst,\ cand,\ 1+i,\ k)$) **endif**
elseif $(n \div 2) < k$ **then** 14
else 13 **endif**

DEFINITION:

```
mjrty-t (a, n, lst)
=   let cand  be  get-nth (0, lst)
    in
    splus (14,
           if (mjrty-k (n, lst, cand, 1, 1) ≃ 0)
              ∨   ((n ÷ 2) < mjrty-k (n, lst, cand, 1, 1))
           then mjrty-sn-t (a, n, lst, cand, 1, 1)
           else splus (mjrty-cand-t (a, n, lst, cand, 1, 1),
                       if cand = mjrty-cand (n, lst, cand, 1, 1)
                       then cand-cnt-t (a,
                                        n,
                                        lst,
                                        mjrty-cand (n,
                                                    lst,
                                                    cand,
                                                    1,
                                                    1),
                                        1,
                                        1)
                       else cand-cnt-t (a,
                                       n,
                                       lst,
                                       mjrty-cand (n,
                                                   lst,
                                                   cand,
                                                   1,
                                                   1),
                                       1,
                                       0) endif) endif) endlet
```

`; induction hints.`

DEFINITION:

```
mjrty-cand-induct (s, n, lst, cand, i, k)
=   if i < n
    then if k ≃ 0
         then let cand1  be  get-nth (i, lst)
              in
              mjrty-cand-induct (stepn (s, 8), n, lst, cand1, 1 + i, 1) endlet
         elseif cand = get-nth (i, lst)
         then mjrty-cand-induct (stepn (s, 9), n, lst, cand, 1 + i, 1 + k)
         else mjrty-cand-induct (stepn (s, 8), n, lst, cand, 1 + i, k − 1) endif
    else t endif
```

DEFINITION:

cand-cnt-induct $(s,\ n,\ lst,\ cand,\ i,\ k)$
$=$ **if** $i < n$
then if $cand = \text{get-nth}\,(i,\ lst)$
then cand-cnt-induct (stepn $(s,\ \mathtt{6}),\ n,\ lst,\ cand,\ 1 + i,\ 1 + k)$
else cand-cnt-induct (stepn $(s,\ \mathtt{5}),\ n,\ lst,\ cand,\ 1 + i,\ k)$ **endif**
else t endif

```
; the preconditions of the initial state.
```

DEFINITION:
mjrty-statep $(s,\ a,\ n,\ lst)$
$=$ ((mc-status (s) = `'running`)
$\land$ evenp (mc-pc (s))
$\land$ rom-addrp (mc-pc (s), mc-mem (s), 130)
$\land$ mcode-addrp (mc-pc (s), mc-mem (s), MJRTY-CODE)
$\land$ ram-addrp (sub (32, 20, read-sp (s)), mc-mem (s), 32)
$\land$ ram-addrp (a, mc-mem (s), $4 * n$)
$\land$ mem-ilst (4, a, mc-mem (s), n, lst)
$\land$ disjoint (a, $4 * n$, sub (32, 20, read-sp (s)), 32)
$\land$ (a = read-mem (add (32, read-sp (s), 4), mc-mem (s), 4))
$\land$ (n = iread-mem (add (32, read-sp (s), 8), mc-mem (s), 4))
$\land$ $(n \not\simeq 0)$)

```
; the conditions of the intermediate state s0.
```

DEFINITION:
mjrty-s0p $(s,\ a,\ n,\ lst,\ cand,\ i,\ k)$
$=$ ((mc-status (s) = `'running`)
$\land$ evenp (mc-pc (s))
$\land$ rom-addrp (sub (32, 50, mc-pc (s)), mc-mem (s), 130)
$\land$ mcode-addrp (sub (32, 50, mc-pc (s)), mc-mem (s), MJRTY-CODE)
$\land$ ram-addrp (sub (32, 16, read-an (32, 6, s)), mc-mem (s), 32)
$\land$ ram-addrp (a, mc-mem (s), $4 * n$)
$\land$ mem-ilst (4, a, mc-mem (s), n, lst)
$\land$ disjoint (a, $4 * n$, sub (32, 16, read-an (32, 6, s)), 32)
$\land$ (a = read-rn (32, 8, mc-rfile (s)))
$\land$ (n = nat-to-int (read-rn (32, 2, mc-rfile (s)), 32))
$\land$ ($cand$ = nat-to-int (read-rn (32, 3, mc-rfile (s)), 32))
$\land$ (i = nat-to-int (read-rn (32, 0, mc-rfile (s)), 32))
$\land$ (k = nat-to-int (read-rn (32, 1, mc-rfile (s)), 32))
$\land$ $(n \not\simeq 0)$
$\land$ $(i \in \mathbf{N})$
$\land$ $(k \in \mathbf{N})$
$\land$ $(k \le i)$)

```
; the conditions of the intermediate state s1.
```

DEFINITION:
mjrty-s1p (s, a, n, lst, $cand$, i, k)
= ((mc-status (s) = 'running)
  ∧ evenp (mc-pc (s))
  ∧ rom-addrp (sub (32, 94, mc-pc (s)), mc-mem (s), 130)
  ∧ mcode-addrp (sub (32, 94, mc-pc (s)), mc-mem (s), MJRTY-CODE)
  ∧ ram-addrp (sub (32, 16, read-an (32, 6, s)), mc-mem (s), 32)
  ∧ ram-addrp (a, mc-mem (s), 4 $*$ n)
  ∧ mem-ilst (4, a, mc-mem (s), n, lst)
  ∧ disjoint (a, 4 $*$ n, sub (32, 16, read-an (32, 6, s)), 32)
  ∧ (a = read-rn (32, 8, mc-rfile (s)))
  ∧ (n = nat-to-int (read-rn (32, 2, mc-rfile (s)), 32))
  ∧ ($cand$ = nat-to-int (read-rn (32, 4, mc-rfile (s)), 32))
  ∧ ($cand$ = nat-to-int (read-rn (32, 3, mc-rfile (s)), 32))
  ∧ (i = nat-to-int (read-rn (32, 0, mc-rfile (s)), 32))
  ∧ (k = nat-to-int (read-rn (32, 1, mc-rfile (s)), 32))
  ∧ ($n \not\simeq 0$)
  ∧ ($i \in \mathbf{N}$)
  ∧ ($k \in \mathbf{N}$)
  ∧ ($k \le i$))

```
; the initial segment.  From the initial state to s0.
```

THEOREM: mjrty-s-s0
let $cand$ **be** get-nth (0, lst)
in
mjrty-statep (s, a, n, lst)
→ (mjrty-s0p (stepn (s, 14), a, n, lst, $cand$, 1, 1)
  ∧ (linked-rts-addr (stepn (s, 14)) = rts-addr (s))
  ∧ (linked-a6 (stepn (s, 14)) = read-an (32, 6, s))
  ∧ (read-rn (32, 14, mc-rfile (stepn (s, 14)))
    = sub (32, 4, read-sp (s)))
  ∧ (movem-saved (stepn (s, 14), 4, 16, 4)
    = readm-rn (32, '(2 3 4 5), mc-rfile (s)))) **endlet**

THEOREM: mjrty-s-s0-rfile
(mjrty-statep (s, a, n, lst) ∧ d6-7a2-5p (rn))
→ (read-rn ($oplen$, rn, mc-rfile (stepn (s, 14)))
  = read-rn ($oplen$, rn, mc-rfile (s)))

THEOREM: mjrty-s-s0-mem
(mjrty-statep (s, a, n, lst) ∧ disjoint (sub (32, 20, read-sp (s)), 32, x, k))
→ (read-mem (x, mc-mem (stepn (s, 14)), k) = read-mem (x, mc-mem (s), k))

```
; s0 --> exit.
; base case.
```

THEOREM: mjrty-s0-sn-base-1
(mjrty-s0p $(s,\ a,\ n,\ lst,\ cand,\ i,\ k) \land (i \not< n) \land (k \simeq \texttt{0})$)
$\rightarrow$ ((mc-status (stepn $(s,\ \texttt{11})$) = `'running`)
  $\land$ (mc-pc (stepn $(s,\ \texttt{11})$) = linked-rts-addr (s))
  $\land$ (iread-dn (32, 0, stepn $(s,\ \texttt{11})$) = $cand$)
  $\land$ (iread-dn (32, 1, stepn $(s,\ \texttt{11})$) = 0)
  $\land$ (read-rn (32, 14, mc-rfile (stepn $(s,\ \texttt{11})$)) = linked-a6 (s))
  $\land$ (read-rn (32, 15, mc-rfile (stepn $(s,\ \texttt{11})$))
    = add (32, read-an (32, 6, s), 8))
  $\land$ (read-mem (x, mc-mem (stepn $(s,\ \texttt{11})$), l)
    = read-mem (x, mc-mem (s), l)))

THEOREM: mjrty-s0-sn-base-2
(mjrty-s0p $(s,\ a,\ n,\ lst,\ cand,\ i,\ k)$
 $\land$ $(i \not< n)$
 $\land$ $(k \not\simeq \texttt{0})$
 $\land$ $((n \div \texttt{2}) < k)$)
$\rightarrow$ ((mc-status (stepn $(s,\ \texttt{17})$) = `'running`)
  $\land$ (mc-pc (stepn $(s,\ \texttt{17})$) = linked-rts-addr (s))
  $\land$ (iread-dn (32, 0, stepn $(s,\ \texttt{17})$) = $cand$)
  $\land$ (iread-dn (32, 1, stepn $(s,\ \texttt{17})$) = 1)
  $\land$ (read-rn (32, 14, mc-rfile (stepn $(s,\ \texttt{17})$)) = linked-a6 (s))
  $\land$ (read-rn (32, 15, mc-rfile (stepn $(s,\ \texttt{17})$))
    = add (32, read-an (32, 6, s), 8))
  $\land$ (read-mem (x, mc-mem (stepn $(s,\ \texttt{17})$), l)
    = read-mem (x, mc-mem (s), l)))

THEOREM: mjrty-s0-sn-rfile-base-1
(mjrty-s0p $(s,\ a,\ n,\ lst,\ cand,\ i,\ k)$
 $\land$ $(i \not< n)$
 $\land$ $(k \simeq \texttt{0})$
 $\land$ d2-7a2-5p (rn)
 $\land$ ($oplen \le$ 32))
$\rightarrow$ (read-rn ($oplen$, rn, mc-rfile (stepn $(s,\ \texttt{11})$))
  = **if** d6-7a2-5p (rn) **then** read-rn ($oplen$, rn, mc-rfile (s))
    else get-vlst ($oplen$,
        0,
        rn,
        '(2 3 4 5),
        movem-saved (s, 4, 16, 4)) **endif**)

THEOREM: mjrty-s0-sn-rfile-base-2

$(\text{mjrty-s0p}\,(s,\ a,\ n,\ lst,\ cand,\ i,\ k)$
$\wedge\quad (i \not< n)$
$\wedge\quad (k \not\simeq 0)$
$\wedge\quad ((n \div 2) < k)$
$\wedge\quad \text{d2-7a2-5p}\,(rn)$
$\wedge\quad (oplen \leq 32))$
$\rightarrow\quad (\text{read-rn}\,(oplen,\ rn,\ \text{mc-rfile}\,(\text{stepn}\,(s,\ 17)))$
$\quad = \quad$ **if** d6-7a2-5p (rn) **then** read-rn $(oplen,\ rn,\ \text{mc-rfile}\,(s))$
else get-vlst $(oplen,$
`0`,
$rn,$
`'(2 3 4 5)`,
movem-saved $(s,\ 4,\ 16,\ 4))$ **endif**)

; induction case.

THEOREM: add1-int-rangep
$(x < \text{nat-to-int}\,(y,\ n)) \rightarrow \text{int-rangep}\,(1 +\ x,\ n)$

EVENT: Enable iplus.

THEOREM: mjrty-s0-s0-1
let *cand1* **be** get-nth $(i,\ lst)$
in
$(\text{mjrty-s0p}\,(s,\ a,\ n,\ lst,\ cand,\ i,\ k) \wedge (i < n) \wedge (k \simeq 0))$
$\rightarrow\quad (\text{mjrty-s0p}\,(\text{stepn}\,(s,\ 8),\ a,\ n,\ lst,\ cand1,\ 1 +\ i,\ 1)$
$\wedge\quad (\text{linked-rts-addr}\,(\text{stepn}\,(s,\ 8)) = \text{linked-rts-addr}\,(s))$
$\wedge\quad (\text{linked-a6}\,(\text{stepn}\,(s,\ 8)) = \text{linked-a6}\,(s))$
$\wedge\quad (\text{read-rn}\,(32,\ 14,\ \text{mc-rfile}\,(\text{stepn}\,(s,\ 8)))$
$\quad = \quad \text{read-rn}\,(32,\ 14,\ \text{mc-rfile}\,(s)))$
$\wedge\quad (\text{movem-saved}\,(\text{stepn}\,(s,\ 8),\ 4,\ 16,\ 4)$
$\quad = \quad \text{movem-saved}\,(s,\ 4,\ 16,\ 4))$
$\wedge\quad (\text{read-mem}\,(x,\ \text{mc-mem}\,(\text{stepn}\,(s,\ 8)),\ l)$
$\quad = \quad \text{read-mem}\,(x,\ \text{mc-mem}\,(s),\ l)))$ **endlet**

THEOREM: add1-int-rangepxx
$((i \leq r) \wedge (r < n) \wedge \text{int-rangep}\,(n,\ 32)) \rightarrow \text{int-rangep}\,(1 +\ i,\ 32)$

THEOREM: mjrty-s0-s0-2
$(\text{mjrty-s0p}\,(s,\ a,\ n,\ lst,\ cand,\ i,\ k)$
$\wedge\quad (i < n)$
$\wedge\quad (k \not\simeq 0)$
$\wedge\quad (cand = \text{get-nth}\,(i,\ lst)))$
$\rightarrow\quad (\text{mjrty-s0p}\,(\text{stepn}\,(s,\ 9),\ a,\ n,\ lst,\ cand,\ 1 +\ i,\ 1 +\ k)$

$$
\begin{array}{l}
\quad\quad \wedge \quad (\text{linked-rts-addr}\,(\text{stepn}\,(s,\ 9)) = \text{linked-rts-addr}\,(s)) \\
\quad\quad \wedge \quad (\text{linked-a6}\,(\text{stepn}\,(s,\ 9)) = \text{linked-a6}\,(s)) \\
\quad\quad \wedge \quad (\text{read-rn}\,(32,\ 14,\ \text{mc-rfile}\,(\text{stepn}\,(s,\ 9))) \\
\quad\quad\quad\quad = \quad \text{read-rn}\,(32,\ 14,\ \text{mc-rfile}\,(s))) \\
\quad\quad \wedge \quad (\text{movem-saved}\,(\text{stepn}\,(s,\ 9),\ 4,\ 16,\ 4) = \text{movem-saved}\,(s,\ 4,\ 16,\ 4)) \\
\quad\quad \wedge \quad (\text{read-mem}\,(x,\ \text{mc-mem}\,(\text{stepn}\,(s,\ 9)),\ l) \\
\quad\quad\quad\quad = \quad \text{read-mem}\,(x,\ \text{mc-mem}\,(s),\ l)))
\end{array}
$$

THEOREM: mjrty-s0-s0-3

$$
\begin{array}{l}
(\text{mjrty-s0p}\,(s,\ a,\ n,\ lst,\ cand,\ i,\ k) \\
\ \wedge \quad (i < n) \\
\ \wedge \quad (k \not\simeq 0) \\
\ \wedge \quad (cand \neq \text{get-nth}\,(i,\ lst))) \\
\rightarrow \quad (\text{mjrty-s0p}\,(\text{stepn}\,(s,\ 8),\ a,\ n,\ lst,\ cand,\ 1 + i,\ k - 1) \\
\quad\quad \wedge \quad (\text{linked-rts-addr}\,(\text{stepn}\,(s,\ 8)) = \text{linked-rts-addr}\,(s)) \\
\quad\quad \wedge \quad (\text{linked-a6}\,(\text{stepn}\,(s,\ 8)) = \text{linked-a6}\,(s)) \\
\quad\quad \wedge \quad (\text{read-rn}\,(32,\ 14,\ \text{mc-rfile}\,(\text{stepn}\,(s,\ 8))) \\
\quad\quad\quad\quad = \quad \text{read-rn}\,(32,\ 14,\ \text{mc-rfile}\,(s))) \\
\quad\quad \wedge \quad (\text{movem-saved}\,(\text{stepn}\,(s,\ 8),\ 4,\ 16,\ 4) = \text{movem-saved}\,(s,\ 4,\ 16,\ 4)) \\
\quad\quad \wedge \quad (\text{read-mem}\,(x,\ \text{mc-mem}\,(\text{stepn}\,(s,\ 8)),\ l) \\
\quad\quad\quad\quad = \quad \text{read-mem}\,(x,\ \text{mc-mem}\,(s),\ l)))
\end{array}
$$

THEOREM: mjrty-s0-s0-rfile-1

$$
\begin{array}{l}
(\text{mjrty-s0p}\,(s,\ a,\ n,\ lst,\ cand,\ i,\ k) \wedge (i < n) \wedge (k \simeq 0) \wedge \text{d6-7a2-5p}\,(rn)) \\
\rightarrow \quad (\text{read-rn}\,(oplen,\ rn,\ \text{mc-rfile}\,(\text{stepn}\,(s,\ 8))) \\
\quad\quad = \quad \text{read-rn}\,(oplen,\ rn,\ \text{mc-rfile}\,(s)))
\end{array}
$$

THEOREM: mjrty-s0-s0-rfile-2

$$
\begin{array}{l}
(\text{mjrty-s0p}\,(s,\ a,\ n,\ lst,\ cand,\ i,\ k) \\
\ \wedge \quad (i < n) \\
\ \wedge \quad (k \not\simeq 0) \\
\ \wedge \quad (cand = \text{get-nth}\,(i,\ lst)) \\
\ \wedge \quad \text{d6-7a2-5p}\,(rn)) \\
\rightarrow \quad (\text{read-rn}\,(oplen,\ rn,\ \text{mc-rfile}\,(\text{stepn}\,(s,\ 9))) \\
\quad\quad = \quad \text{read-rn}\,(oplen,\ rn,\ \text{mc-rfile}\,(s)))
\end{array}
$$

THEOREM: mjrty-s0-s0-rfile-3

$$
\begin{array}{l}
(\text{mjrty-s0p}\,(s,\ a,\ n,\ lst,\ cand,\ i,\ k) \\
\ \wedge \quad (i < n) \\
\ \wedge \quad (k \not\simeq 0) \\
\ \wedge \quad (cand \neq \text{get-nth}\,(i,\ lst)) \\
\ \wedge \quad \text{d6-7a2-5p}\,(rn)) \\
\rightarrow \quad (\text{read-rn}\,(oplen,\ rn,\ \text{mc-rfile}\,(\text{stepn}\,(s,\ 8))) \\
\quad\quad = \quad \text{read-rn}\,(oplen,\ rn,\ \text{mc-rfile}\,(s)))
\end{array}
$$

```
; the proof of s0 --> exit.
```

THEOREM: mjrty-s0-sn

$$
\begin{array}{l}
\textbf{let } \mathit{sn} \textbf{ be } \text{stepn}(s, \text{mjrty-sn-t}(a, n, \mathit{lst}, \mathit{cand}, i, k)) \\
\textbf{in} \\
(\text{mjrty-s0p}(s, a, n, \mathit{lst}, \mathit{cand}, i, k) \\
\quad \land \quad ((\text{mjrty-k}(n, \mathit{lst}, \mathit{cand}, i, k) \simeq \mathtt{0}) \\
\qquad\quad \lor \quad ((n \div \mathtt{2}) < \text{mjrty-k}(n, \mathit{lst}, \mathit{cand}, i, k)))) \\
\rightarrow \quad ((\text{mc-status}(\mathit{sn}) = \mathtt{'running}) \\
\qquad\quad \land \quad (\text{mc-pc}(\mathit{sn}) = \text{linked-rts-addr}(s)) \\
\qquad\quad \land \quad (\text{iread-dn}(\mathtt{32}, \mathtt{0}, \mathit{sn}) = \text{mjrty-cand}(n, \mathit{lst}, \mathit{cand}, i, k)) \\
\qquad\quad \land \quad (\text{iread-dn}(\mathtt{32}, \mathtt{1}, \mathit{sn}) \\
\qquad\qquad\quad = \quad \textbf{if } (n \div \mathtt{2}) < \text{mjrty-k}(n, \mathit{lst}, \mathit{cand}, i, k) \\
\qquad\qquad\qquad\quad \textbf{then } \mathtt{1} \\
\qquad\qquad\qquad\quad \textbf{else } \mathtt{0} \textbf{ endif}) \\
\qquad\quad \land \quad (\text{read-rn}(\mathtt{32}, \mathtt{14}, \text{mc-rfile}(\mathit{sn})) = \text{linked-a6}(s)) \\
\qquad\quad \land \quad (\text{read-rn}(\mathtt{32}, \mathtt{15}, \text{mc-rfile}(\mathit{sn})) \\
\qquad\qquad\quad = \quad \text{add}(\mathtt{32}, \text{read-an}(\mathtt{32}, \mathtt{6}, s), \mathtt{8})) \\
\qquad\quad \land \quad (\text{read-mem}(x, \text{mc-mem}(\mathit{sn}), l) = \text{read-mem}(x, \text{mc-mem}(s), l))) \textbf{ endlet}
\end{array}
$$

THEOREM: mjrty-s0-sn-rfile

$$
\begin{array}{l}
\textbf{let } \mathit{sn} \textbf{ be } \text{stepn}(s, \text{mjrty-sn-t}(a, n, \mathit{lst}, \mathit{cand}, i, k)) \\
\textbf{in} \\
(\text{mjrty-s0p}(s, a, n, \mathit{lst}, \mathit{cand}, i, k) \\
\quad \land \quad ((\text{mjrty-k}(n, \mathit{lst}, \mathit{cand}, i, k) \simeq \mathtt{0}) \\
\qquad\quad \lor \quad ((n \div \mathtt{2}) < \text{mjrty-k}(n, \mathit{lst}, \mathit{cand}, i, k))) \\
\quad \land \quad \text{d2-7a2-5p}(\mathit{rn}) \\
\quad \land \quad (\mathit{oplen} \le \mathtt{32})) \\
\rightarrow \quad (\text{read-rn}(\mathit{oplen}, \mathit{rn}, \text{mc-rfile}(\mathit{sn})) \\
\qquad = \quad \textbf{if } \text{d6-7a2-5p}(\mathit{rn}) \textbf{ then } \text{read-rn}(\mathit{oplen}, \mathit{rn}, \text{mc-rfile}(s)) \\
\qquad\qquad \textbf{else } \text{get-vlst}(\mathit{oplen}, \\
\qquad\qquad\qquad\qquad\quad \mathtt{0}, \\
\qquad\qquad\qquad\qquad\quad \mathit{rn}, \\
\qquad\qquad\qquad\qquad\quad \mathtt{'(2\ 3\ 4\ 5)}, \\
\qquad\qquad\qquad\qquad\quad \text{movem-saved}(s, \mathtt{4}, \mathtt{16}, \mathtt{4})) \textbf{ endif}) \textbf{ endlet}
\end{array}
$$

```
; s0 --> s1.
; base case:
```

THEOREM: mjrty-s0-s1-base1

$$
\begin{array}{l}
(\text{mjrty-s0p}(s, a, n, \mathit{lst}, \mathit{cand}, i, k) \\
\quad \land \quad (i \not< n) \\
\quad \land \quad (k \not\simeq \mathtt{0}) \\
\quad \land \quad ((n \div \mathtt{2}) \not< k) \\
\quad \land \quad (\mathit{cand} = \text{get-nth}(\mathtt{0}, \mathit{lst})))
\end{array}
$$

$$\begin{array}{ll}\rightarrow & (\text{mjrty-s1p}\,(\text{stepn}\,(s,\ \mathtt{18}),\ a,\ n,\ \mathit{lst},\ \mathit{cand},\ \mathtt{1},\ \mathtt{1}) \\ & \quad \wedge\ \ (\text{linked-rts-addr}\,(\text{stepn}\,(s,\ \mathtt{18})) = \text{linked-rts-addr}\,(s)) \\ & \quad \wedge\ \ (\text{linked-a6}\,(\text{stepn}\,(s,\ \mathtt{18})) = \text{linked-a6}\,(s)) \\ & \quad \wedge\ \ (\text{read-rn}\,(\mathtt{32},\ \mathtt{14},\ \text{mc-rfile}\,(\text{stepn}\,(s,\ \mathtt{18}))) \\ & \qquad\quad = \ \ \text{read-rn}\,(\mathtt{32},\ \mathtt{14},\ \text{mc-rfile}\,(s))) \\ & \quad \wedge\ \ (\text{movem-saved}\,(\text{stepn}\,(s,\ \mathtt{18}),\ \mathtt{4},\ \mathtt{16},\ \mathtt{4}) = \text{movem-saved}\,(s,\ \mathtt{4},\ \mathtt{16},\ \mathtt{4})) \\ & \quad \wedge\ \ (\text{read-mem}\,(x,\ \text{mc-mem}\,(\text{stepn}\,(s,\ \mathtt{18})),\ l) \\ & \qquad\quad = \ \ \text{read-mem}\,(x,\ \text{mc-mem}\,(s),\ l)))\end{array}$$

Theorem: mjrty-s0-s1-base2

$$\begin{array}{ll} & (\text{mjrty-s0p}\,(s,\ a,\ n,\ \mathit{lst},\ \mathit{cand},\ i,\ k) \\ \wedge & (i \not< n) \\ \wedge & (k \not\simeq \mathtt{0}) \\ \wedge & ((n \div \mathtt{2}) \not< k) \\ \wedge & (\mathit{cand} \neq \text{get-nth}\,(\mathtt{0},\ \mathit{lst}))) \\ \rightarrow & (\text{mjrty-s1p}\,(\text{stepn}\,(s,\ \mathtt{17}),\ a,\ n,\ \mathit{lst},\ \mathit{cand},\ \mathtt{1},\ \mathtt{0}) \\ & \quad \wedge\ \ (\text{linked-rts-addr}\,(\text{stepn}\,(s,\ \mathtt{17})) = \text{linked-rts-addr}\,(s)) \\ & \quad \wedge\ \ (\text{linked-a6}\,(\text{stepn}\,(s,\ \mathtt{17})) = \text{linked-a6}\,(s)) \\ & \quad \wedge\ \ (\text{read-rn}\,(\mathtt{32},\ \mathtt{14},\ \text{mc-rfile}\,(\text{stepn}\,(s,\ \mathtt{17}))) \\ & \qquad\quad = \ \ \text{read-rn}\,(\mathtt{32},\ \mathtt{14},\ \text{mc-rfile}\,(s))) \\ & \quad \wedge\ \ (\text{movem-saved}\,(\text{stepn}\,(s,\ \mathtt{17}),\ \mathtt{4},\ \mathtt{16},\ \mathtt{4}) = \text{movem-saved}\,(s,\ \mathtt{4},\ \mathtt{16},\ \mathtt{4})) \\ & \quad \wedge\ \ (\text{read-mem}\,(x,\ \text{mc-mem}\,(\text{stepn}\,(s,\ \mathtt{17})),\ l) \\ & \qquad\quad = \ \ \text{read-mem}\,(x,\ \text{mc-mem}\,(s),\ l)))\end{array}$$

Theorem: mjrty-s0-s1-rfile-base1

$$\begin{array}{ll} & (\text{mjrty-s0p}\,(s,\ a,\ n,\ \mathit{lst},\ \mathit{cand},\ i,\ k) \\ \wedge & (i \not< n) \\ \wedge & (k \not\simeq \mathtt{0}) \\ \wedge & ((n \div \mathtt{2}) \not< k) \\ \wedge & (\mathit{cand} = \text{get-nth}\,(\mathtt{0},\ \mathit{lst})) \\ \wedge & \text{d6-7a2-5p}\,(\mathit{rn})) \\ \rightarrow & (\text{read-rn}\,(\mathit{oplen},\ \mathit{rn},\ \text{mc-rfile}\,(\text{stepn}\,(s,\ \mathtt{18}))) \\ & \quad = \ \ \text{read-rn}\,(\mathit{oplen},\ \mathit{rn},\ \text{mc-rfile}\,(s)))\end{array}$$

Theorem: mjrty-s0-s1-rfile-base2

$$\begin{array}{ll} & (\text{mjrty-s0p}\,(s,\ a,\ n,\ \mathit{lst},\ \mathit{cand},\ i,\ k) \\ \wedge & (i \not< n) \\ \wedge & (k \not\simeq \mathtt{0}) \\ \wedge & ((n \div \mathtt{2}) \not< k) \\ \wedge & (\mathit{cand} \neq \text{get-nth}\,(\mathtt{0},\ \mathit{lst})) \\ \wedge & \text{d6-7a2-5p}\,(\mathit{rn})) \\ \rightarrow & (\text{read-rn}\,(\mathit{oplen},\ \mathit{rn},\ \text{mc-rfile}\,(\text{stepn}\,(s,\ \mathtt{17}))) \\ & \quad = \ \ \text{read-rn}\,(\mathit{oplen},\ \mathit{rn},\ \text{mc-rfile}\,(s)))\end{array}$$

```
; the proof of s0 --> s1.
```

THEOREM: mjrty-s0-s1

$$
\begin{array}{l}
\textbf{let } \mathit{s1} \textbf{ be } \text{stepn}(s, \text{mjrty-cand-t}(a, n, \mathit{lst}, \mathit{cand}, i, k)) \\
\textbf{in} \\
(\text{mjrty-s0p}(s, a, n, \mathit{lst}, \mathit{cand}, i, k) \\
\ \wedge \quad (\text{mjrty-k}(n, \mathit{lst}, \mathit{cand}, i, k) \not\simeq \texttt{0}) \\
\ \wedge \quad ((n \div \texttt{2}) \not< \text{mjrty-k}(n, \mathit{lst}, \mathit{cand}, i, k)) \\
\ \wedge \quad (\mathit{cand0} = \text{mjrty-cand}(n, \mathit{lst}, \mathit{cand}, i, k)) \\
\ \wedge \quad (\mathit{k0} = \textbf{if } \text{mjrty-cand}(n, \mathit{lst}, \mathit{cand}, i, k) = \text{get-nth}(\texttt{0}, \mathit{lst}) \\
\qquad\qquad \textbf{then } \texttt{1} \\
\qquad\qquad \textbf{else } \texttt{0} \textbf{ endif})) \\
\to \quad (\text{mjrty-s1p}(\mathit{s1}, a, n, \mathit{lst}, \mathit{cand0}, \texttt{1}, \mathit{k0}) \\
\qquad \wedge \quad (\text{linked-rts-addr}(\mathit{s1}) = \text{linked-rts-addr}(s)) \\
\qquad \wedge \quad (\text{linked-a6}(\mathit{s1}) = \text{linked-a6}(s)) \\
\qquad \wedge \quad (\text{read-rn}(\texttt{32}, \texttt{14}, \text{mc-rfile}(\mathit{s1})) \\
\qquad\qquad = \quad \text{read-rn}(\texttt{32}, \texttt{14}, \text{mc-rfile}(s))) \\
\qquad \wedge \quad (\text{movem-saved}(\mathit{s1}, \texttt{4}, \texttt{16}, \texttt{4}) = \text{movem-saved}(s, \texttt{4}, \texttt{16}, \texttt{4})) \\
\qquad \wedge \quad (\text{read-mem}(x, \text{mc-mem}(\mathit{s1}), l) = \text{read-mem}(x, \text{mc-mem}(s), l))) \textbf{ endlet}
\end{array}
$$

THEOREM: mjrty-s0-s1-rfile

$$
\begin{array}{l}
\textbf{let } \mathit{s1} \textbf{ be } \text{stepn}(s, \text{mjrty-cand-t}(a, n, \mathit{lst}, \mathit{cand}, i, k)) \\
\textbf{in} \\
(\text{mjrty-s0p}(s, a, n, \mathit{lst}, \mathit{cand}, i, k) \\
\ \wedge \quad (\text{mjrty-k}(n, \mathit{lst}, \mathit{cand}, i, k) \not\simeq \texttt{0}) \\
\ \wedge \quad ((n \div \texttt{2}) \not< \text{mjrty-k}(n, \mathit{lst}, \mathit{cand}, i, k)) \\
\ \wedge \quad \text{d6-7a2-5p}(\mathit{rn})) \\
\to \quad (\text{read-rn}(\mathit{oplen}, \mathit{rn}, \text{mc-rfile}(\mathit{s1})) \\
\qquad = \quad \text{read-rn}(\mathit{oplen}, \mathit{rn}, \text{mc-rfile}(s))) \textbf{ endlet}
\end{array}
$$

```
; s1 --> exit.
; base case.
```

THEOREM: mjrty-s1-sn-1

$$
\begin{array}{l}
(\text{mjrty-s1p}(s, a, n, \mathit{lst}, \mathit{cand}, i, k) \wedge (i \not< n) \wedge ((n \div \texttt{2}) < k)) \\
\to \quad ((\text{mc-status}(\text{stepn}(s, \texttt{14})) = \texttt{'running}) \\
\qquad \wedge \quad (\text{mc-pc}(\text{stepn}(s, \texttt{14})) = \text{linked-rts-addr}(s)) \\
\qquad \wedge \quad (\text{iread-dn}(\texttt{32}, \texttt{0}, \text{stepn}(s, \texttt{14})) = \mathit{cand}) \\
\qquad \wedge \quad (\text{iread-dn}(\texttt{32}, \texttt{1}, \text{stepn}(s, \texttt{14})) = \texttt{1}) \\
\qquad \wedge \quad (\text{read-rn}(\texttt{32}, \texttt{14}, \text{mc-rfile}(\text{stepn}(s, \texttt{14}))) = \text{linked-a6}(s)) \\
\qquad \wedge \quad (\text{read-rn}(\texttt{32}, \texttt{15}, \text{mc-rfile}(\text{stepn}(s, \texttt{14}))) \\
\qquad\qquad = \quad \text{add}(\texttt{32}, \text{read-an}(\texttt{32}, \texttt{6}, s), \texttt{8})) \\
\qquad \wedge \quad (\text{read-mem}(x, \text{mc-mem}(\text{stepn}(s, \texttt{14})), l) \\
\qquad\qquad = \quad \text{read-mem}(x, \text{mc-mem}(s), l)))
\end{array}
$$

THEOREM: mjrty-s1-sn-rfile-1

$$
(\text{mjrty-s1p}(s, a, n, \mathit{lst}, \mathit{cand}, i, k)
$$

$\wedge \quad (i \not< n)$
$\wedge \quad ((n \div 2) < k)$
$\wedge \quad$ d2-7a2-5p (rn)
$\wedge \quad (oplen \leq 32))$
$\rightarrow \quad$ (read-rn $(oplen,\ rn,$ mc-rfile (stepn $(s,\ 14)))$
$\quad = \quad$ **if** d6-7a2-5p (rn) **then** read-rn $(oplen,\ rn,$ mc-rfile $(s))$
else get-vlst $(oplen,$
`0`,
$rn,$
`'(2 3 4 5)`,
movem-saved $(s,\ 4,\ 16,\ 4))$ **endif**)

THEOREM: mjrty-s1-sn-2
(mjrty-s1p $(s,\ a,\ n,\ lst,\ cand,\ i,\ k) \wedge (i \not< n) \wedge ((n \div 2) \not< k))$
$\rightarrow \quad$ ((mc-status (stepn $(s,\ 13)) =$ `'running`)
$\wedge \quad$ (mc-pc (stepn $(s,\ 13)) =$ linked-rts-addr $(s))$
$\wedge \quad$ (iread-dn $(32,\ 0,$ stepn $(s,\ 13)) = cand)$
$\wedge \quad$ (iread-dn $(32,\ 1,$ stepn $(s,\ 13)) = 0)$
$\wedge \quad$ (read-rn $(32,\ 14,$ mc-rfile (stepn $(s,\ 13))) =$ linked-a6 $(s))$
$\wedge \quad$ (read-rn $(32,\ 15,$ mc-rfile (stepn $(s,\ 13)))$
$\quad = \quad$ add $(32,$ read-an $(32,\ 6,\ s),\ 8))$
$\wedge \quad$ (read-mem $(x,$ mc-mem (stepn $(s,\ 13)),\ l)$
$\quad = \quad$ read-mem $(x,$ mc-mem $(s),\ l)))$

THEOREM: mjrty-s1-sn-rfile-2
(mjrty-s1p $(s,\ a,\ n,\ lst,\ cand,\ i,\ k)$
$\wedge \quad (i \not< n)$
$\wedge \quad ((n \div 2) \not< k)$
$\wedge \quad$ d2-7a2-5p (rn)
$\wedge \quad (oplen \leq 32))$
$\rightarrow \quad$ (read-rn $(oplen,\ rn,$ mc-rfile (stepn $(s,\ 13)))$
$\quad = \quad$ **if** d6-7a2-5p (rn) **then** read-rn $(oplen,\ rn,$ mc-rfile $(s))$
else get-vlst $(oplen,$
`0`,
$rn,$
`'(2 3 4 5)`,
movem-saved $(s,\ 4,\ 16,\ 4))$ **endif**)

`; induction case.`

THEOREM: mjrty-s1-s1-1
(mjrty-s1p $(s,\ a,\ n,\ lst,\ cand,\ i,\ k) \wedge (i < n) \wedge (cand =$ get-nth $(i,\ lst)))$
$\rightarrow \quad$ (mjrty-s1p (stepn $(s,\ 6),\ a,\ n,\ lst,\ cand,\ 1 + i,\ 1 + k)$
$\wedge \quad$ (linked-rts-addr (stepn $(s,\ 6)) =$ linked-rts-addr $(s))$
$\wedge \quad$ (linked-a6 (stepn $(s,\ 6)) =$ linked-a6 $(s))$

$\wedge$ (read-rn (32, 14, mc-rfile (stepn (s, 6)))
$\quad = \quad$ read-rn (32, 14, mc-rfile (s)))
$\wedge$ (movem-saved (stepn (s, 6), 4, 16, 4) = movem-saved (s, 4, 16, 4))
$\wedge$ (read-mem (x, mc-mem (stepn (s, 6)), l)
$\quad = \quad$ read-mem (x, mc-mem (s), l)))

THEOREM: mjrty-s1-s1-2
(mjrty-s1p (s, a, n, lst, $cand$, i, k) $\wedge$ ($i < n$) $\wedge$ ($cand \neq$ get-nth (i, lst)))
$\rightarrow$ (mjrty-s1p (stepn (s, 5), a, n, lst, $cand$, $1 + i$, k)
$\wedge$ (linked-rts-addr (stepn (s, 5)) = linked-rts-addr (s))
$\wedge$ (linked-a6 (stepn (s, 5)) = linked-a6 (s))
$\wedge$ (read-rn (32, 14, mc-rfile (stepn (s, 5)))
$\quad = \quad$ read-rn (32, 14, mc-rfile (s)))
$\wedge$ (movem-saved (stepn (s, 5), 4, 16, 4) = movem-saved (s, 4, 16, 4))
$\wedge$ (read-mem (x, mc-mem (stepn (s, 5)), l)
$\quad = \quad$ read-mem (x, mc-mem (s), l)))

THEOREM: mjrty-s1-s1-rfile-1
(mjrty-s1p (s, a, n, lst, $cand$, i, k)
$\wedge$ ($i < n$)
$\wedge$ ($cand$ = get-nth (i, lst))
$\wedge$ d6-7a2-5p (rn))
$\rightarrow$ (read-rn ($oplen$, rn, mc-rfile (stepn (s, 6)))
$\quad = \quad$ read-rn ($oplen$, rn, mc-rfile (s)))

THEOREM: mjrty-s1-s1-rfile-2
(mjrty-s1p (s, a, n, lst, $cand$, i, k)
$\wedge$ ($i < n$)
$\wedge$ ($cand \neq$ get-nth (i, lst))
$\wedge$ d6-7a2-5p (rn))
$\rightarrow$ (read-rn ($oplen$, rn, mc-rfile (stepn (s, 5)))
$\quad = \quad$ read-rn ($oplen$, rn, mc-rfile (s)))

THEOREM: mjrty-s1-sn
let sn **be** stepn (s, cand-cnt-t (a, n, lst, $cand$, i, k))
in
mjrty-s1p (s, a, n, lst, $cand$, i, k)
$\rightarrow$ ((mc-status (sn) = `'running`)
$\wedge$ (mc-pc (sn) = linked-rts-addr (s))
$\wedge$ (iread-dn (32, 0, sn) = $cand$)
$\wedge$ (iread-dn (32, 1, sn)
$\quad = \quad$ **if** ($n \div 2$) < cand-cnt (n, lst, $cand$, i, k)
then 1
else 0 **endif**)
$\wedge$ (read-rn (32, 14, mc-rfile (sn)) = linked-a6 (s))

$\wedge$ (read-rn (32, 15, mc-rfile (sn))
= add (32, read-an (32, 6, s), 8))
$\wedge$ (read-mem (x, mc-mem (sn), l) = read-mem (x, mc-mem (s), l))) **endlet**

THEOREM: mjrty-s1-sn-rfile
let sn **be** stepn (s, cand-cnt-t (a, n, lst, $cand$, i, k))
in
(mjrty-s1p (s, a, n, lst, $cand$, i, k) $\wedge$ d2-7a2-5p (rn) $\wedge$ ($oplen \leq$ 32))
$\rightarrow$ (read-rn ($oplen$, rn, mc-rfile (sn))
= **if** d6-7a2-5p (rn) **then** read-rn ($oplen$, rn, mc-rfile (s))
else get-vlst ($oplen$,
0,
rn,
'(2 3 4 5),
movem-saved (s, 4, 16, 4)) **endif**) **endlet**

`; the correctness of MJRTY.`

THEOREM: mjrty-statep-info
mjrty-statep (s, a, n, lst) $\rightarrow$ ($n \not\simeq$ 0)

THEOREM: mjrty-correctness
let sn **be** stepn (s, mjrty-t (a, n, lst))
in
mjrty-statep (s, a, n, lst)
$\rightarrow$ ((mc-status (sn) = '**running**)
$\wedge$ (mc-pc (sn) = rts-addr (s))
$\wedge$ (read-rn (32, 14, mc-rfile (sn))
= read-rn (32, 14, mc-rfile (s)))
$\wedge$ (read-rn (32, 15, mc-rfile (sn))
= add (32, read-sp (s), 4))
$\wedge$ ((d2-7a2-5p (rn) $\wedge$ ($oplen \leq$ 32))
$\rightarrow$ (read-rn ($oplen$, rn, mc-rfile (sn))
= read-rn ($oplen$, rn, mc-rfile (s))))
$\wedge$ (disjoint (sub (32, 20, read-sp (s)), 32, x, k)
$\rightarrow$ (read-mem (x, mc-mem (sn), k)
= read-mem (x, mc-mem (s), k)))
$\wedge$ (iread-dn (32, 0, sn) = mjrty-cand (n, lst, 0, 0, 0))
$\wedge$ (iread-dn (32, 1, sn)
= **if** mjrty-p (n, lst, 0, 0, 0) **then** 1
else 0 **endif**)) **endlet**

EVENT: Disable mjrty-t.

`; in the logic, mjrty is expected to have these properties:`

```
;    1.  mjrty-thm-1: if mjrty-p returns 1,  cand wins the majority.
;    2.  mjrty-thm-2: if mjrty-p returns 0,  no one wins the majority.
```

THEOREM: mjrty-cand-0
mjrty-cand $(n,\ lst,\ x,\ n,\ k) = x$

THEOREM: mjrty-cand-1
mjrty-cand $(1 + n,\ lst,\ x,\ n,\ k)$
$=$ **if** $k \simeq 0$ **then** get-nth $(n,\ lst)$
else x **endif**

THEOREM: mjrty-k-0
mjrty-k $(n,\ lst,\ x,\ n,\ k) = k$

THEOREM: mjrty-k-1
mjrty-k $(1 + n,\ lst,\ x,\ n,\ k)$
$=$ **if** $k \simeq 0$ **then** 1
elseif $x =$ get-nth $(n,\ lst)$ **then** $1 + k$
else $k - 1$ **endif**

THEOREM: cand-cnt-0
cand-cnt $(n,\ lst,\ x,\ n,\ k) = k$

THEOREM: cand-cnt-1
cand-cnt $(1 + n,\ lst,\ x,\ n,\ k)$
$=$ **if** $x =$ get-nth $(n,\ lst)$ **then** $1 + k$
else k **endif**

THEOREM: mjrty-k-lemma
$((i \le n) \wedge (j \le i) \wedge (i \in \mathbf{N}))$
$\rightarrow$ (mjrty-k $(n,\ lst,$ mjrty-cand $(i,\ lst,\ x,\ j,\ k),\ i,$ mjrty-k $(i,\ lst,\ x,\ j,\ k))$
$=$ mjrty-k $(n,\ lst,\ x,\ j,\ k))$

THEOREM: mjrty-cand-lemma
$((i \le n) \wedge (j \le i) \wedge (i \in \mathbf{N}))$
$\rightarrow$ (mjrty-cand $(n,\ lst,$ mjrty-cand $(i,\ lst,\ x,\ j,\ k),\ i,$ mjrty-k $(i,\ lst,\ x,\ j,\ k))$
$=$ mjrty-cand $(n,\ lst,\ x,\ j,\ k))$

THEOREM: cand-cnt-lemma
$((i \le n) \wedge (j \le i) \wedge (i \in \mathbf{N}))$
$\rightarrow$ (cand-cnt $(n,\ lst,\ x,\ i,$ cand-cnt $(i,\ lst,\ x,\ j,\ k)) =$ cand-cnt $(n,\ lst,\ x,\ j,\ k))$

THEOREM: mjrty-cand-rec
$(n \in \mathbf{N})$
$\rightarrow$ (mjrty-cand $(1 + n,\ lst,\ x,\ 0,\ 0)$
$=$ mjrty-cand $(1 + n,$

lst,
mjrty-cand $(n,\ lst,\ x,\ 0,\ 0)$,
n,
mjrty-k $(n,\ lst,\ x,\ 0,\ 0)))$

THEOREM: mjrty-k-rec
$(n \in \mathbf{N})$
$\rightarrow$ (mjrty-k $(1 +\ n,\ lst,\ x,\ 0,\ 0)$
$=$ mjrty-k $(1 +\ n,$
lst,
mjrty-cand $(n,\ lst,\ x,\ 0,\ 0)$,
n,
mjrty-k $(n,\ lst,\ x,\ 0,\ 0)))$

THEOREM: cand-cnt-rec
$(n \in \mathbf{N})$
$\rightarrow$ (cand-cnt $(1 +\ n,\ lst,\ x,\ 0,\ 0)$
$=$ cand-cnt $(1 +\ n,\ lst,\ x,\ n,$ cand-cnt $(n,\ lst,\ x,\ 0,\ 0)))$

EVENT: Disable mjrty-cand-lemma.

EVENT: Disable mjrty-k-lemma.

EVENT: Disable cand-cnt-lemma.

THEOREM: mjrty-lemma1
cand-cnt $(n,\ lst,$ mjrty-cand $(n,\ lst,\ 0,\ 0,\ 0),\ 0,\ 0) \not<$ mjrty-k $(n,\ lst,\ 0,\ 0,\ 0)$

DEFINITION:
mjrty-lemma2-induct $(n,\ lst,\ x)$
$=$ **if** $n \simeq 0$ **then t**
else mjrty-lemma2-induct $(n - 1,\ lst,\ x)$
$\wedge$ mjrty-lemma2-induct $(n - 1,\ lst,$ get-nth $(n - 1,\ lst))$ **endif**

THEOREM: mjrty-lemma2
$((n +$ mjrty-k $(n,\ lst,\ 0,\ 0,\ 0))$
$\not<$ $(2 *$ cand-cnt $(n,\ lst,$ mjrty-cand $(n,\ lst,\ 0,\ 0,\ 0),\ 0,\ 0)))$
$\wedge$ $((x \neq$ mjrty-cand $(n,\ lst,\ 0,\ 0,\ 0))$
$\rightarrow$ $(n \not<$ (mjrty-k $(n,\ lst,\ 0,\ 0,\ 0)$
$+$ $(2 *$ cand-cnt $(n,\ lst,\ x,\ 0,\ 0)))))$

EVENT: Disable mjrty-cand-rec.

EVENT: Disable mjrty-k-rec.

EVENT: Disable cand-cnt-rec.

THEOREM: mjrty-thm1
$\text{mjrty-p}\,(n,\ lst,\ 0,\ 0,\ 0)$
$\rightarrow\ \ ((n \div 2) < \text{cand-cnt}\,(n,\ lst,\ \text{mjrty-cand}\,(n,\ lst,\ 0,\ 0,\ 0),\ 0,\ 0))$

THEOREM: mjrty-thm2
$(\neg\ \text{mjrty-p}\,(n,\ lst,\ 0,\ 0,\ 0)) \rightarrow ((n \div 2) \not< \text{cand-cnt}\,(n,\ lst,\ x,\ 0,\ 0))$

```
; a simple time analysis.
```

THEOREM: mjrty-t-crock
$(z * ((x - 1) - y)) = ((z * (x - y)) - z)$

THEOREM: mjrty-cand-t-0
(mjrty-cand-t $(a,\ 0,\ lst,\ cand,\ i,\ k)$
  = **if** $cand = \text{get-nth}\,(0,\ lst)$ **then** 18
    else 17 **endif**)
$\wedge$ (mjrty-cand-t $(a,\ 1,\ lst,\ cand,\ 1,\ k)$
    = **if** $cand = \text{get-nth}\,(0,\ lst)$ **then** 18
      else 17 **endif**)

THEOREM: mjrty-cand-t-1
mjrty-cand-t $(a,\ 1,\ lst,\ cand,\ i,\ k)$
= **if** $i \simeq 0$
  then if $k \simeq 0$ **then** 26
    elseif $cand = \text{get-nth}\,(i,\ lst)$
    then if $cand = \text{get-nth}\,(0,\ lst)$ **then** 27
      else 26 **endif**
    elseif $cand = \text{get-nth}\,(0,\ lst)$ **then** 26
    else 25 **endif**
  elseif $cand = \text{get-nth}\,(0,\ lst)$ **then** 18
  else 17 **endif**

THEOREM: mjrty-cand-t-ubound
$(18 + (9 * (n - i))) \not< \text{mjrty-cand-t}\,(a,\ n,\ lst,\ cand,\ i,\ k)$

THEOREM: mjrty-sn-t-ubound
$(17 + (9 * (n - i))) \not< \text{mjrty-sn-t}\,(a,\ n,\ lst,\ cand,\ i,\ k)$

THEOREM: cand-cnt-t-0
(cand-cnt-t $(a,\ 0,\ lst,\ cand,\ i,\ k)$
  = **if** $0 < k$ **then** 14
    else 13 **endif**)
$\wedge$ (cand-cnt-t $(a,\ n,\ lst,\ cand,\ n,\ k)$
    = **if** $(n \div 2) < k$ **then** 14
      else 13 **endif**)

THEOREM: cand-cnt-t-1
cand-cnt-t$(a,\ 1,\ lst,\ cand,\ i,\ k)$
$=$ **if** $i \simeq 0$
then if $cand =$ get-nth$(i,\ lst)$ **then** 20
elseif $0 < k$ **then** 19
else 18 **endif**
elseif $0 < k$ **then** 14
else 13 **endif**

THEOREM: cand-cnt-t-ubound
$(14 + (6 * (n - i))) \not< $ cand-cnt-t$(a,\ n,\ lst,\ cand,\ i,\ k)$

THEOREM: mjrty-t-ubound
mjrty-t$(a,\ n,\ lst) \leq (46 + (15 * (n - 1)))$

Index