The Boyer-Moore Majority Vote Algorithm

with a majority of voting rabbits

Wim H. Hesselink, 7th November 2005

The majority vote problem is to determine in any given sequence of votes whether there is a candidate with more votes than all the others, and if so, to determine this candidate. The Boyer-Moore majority vote algorithm solves the problem in time linear in the length of the sequence and constant memory.

It does so in two repetitions. The first repetition eliminates all candidates but one. The second repetition verifies whether or not the remaining candidate holds a majority. Given the postcondition of the first repetition, it is easy to implement the second repetition, and to verify its correctness.

The first repetition is more interesting. In this note we therefore try to develop the first repetition from its postcondition. We model the sequence of votes by an array a, and the candidate to be chosen by a variable p, according to the following declaration.

```
const $n : \mathit{Nat}$ ;
   $a :$ array $[0 \ldots n)$ of $\mathit{Candidate}$ ;
var $p : \mathit{Candidate}$ .
```

The postcondition of the first repetition is that every candidate other than p does not hold a majority:

$$Q : \quad (\forall\, q : q \neq p \;\Rightarrow\; \#\{i \mid a[i] = q\} \leq \tfrac{1}{2} \cdot n) \ .$$

In order to establish this postcondition, it is natural to inspect the elements of array a one by one, and to introduce a loop variable

```
var $k : \mathit{Nat}$ .
```

We now define $\mathit{votes}(k, q) = \#\{i \mid i < k \wedge a[i] = q\}$. Then we have

$$Q \quad \equiv \quad (\forall\, q : q \neq p \;\Rightarrow\; 2 \cdot \mathit{votes}(n, q) \leq n) \ .$$

This suggests the generalization

$$P : \quad (\forall\, q : q \neq p \;\Rightarrow\; 2 \cdot \mathit{votes}(k, q) \leq k) \ .$$

It is clear that Q follows from $P \wedge k = n$. On the other hand, P holds trivially for $k = 0$. When P holds and testing yields $a[k] = p$, we can increment k and sharpen the majority estimates since the values $\mathit{votes}(k, q)$ remain unchanged for $q \neq p$. We therefore introduce a variable s to indicate the sharpening:

```
var $s : \mathit{Int}$ .
```

The sharpening is expressed in the proposed invariant

$$J0 : \quad s \geq 0 \;\wedge\; k \leq n \;\wedge\; (\forall\, q : q \neq p \;\Rightarrow\; s + 2 \cdot \mathit{votes}(k, q) \leq k) \ .$$

We now choose $B : k \neq n$ as guard of the repetition to be developed. It is easy to verify that $[J0 \wedge \neg B \;\Rightarrow\; Q]$.

In the following analysis, we use the terms incrementation and decrementation to mean incrementation and decrementation with 1.

When $J0 \wedge k \neq n$ holds and $a[k] = p$, then $J0$ is preserved when s and k are incremented. When $J0 \wedge k \neq n$ holds and $s > 0$ (and $a[k] \neq p$), then $J0$ is preserved when k is incremented and s is decremented. When $J0 \wedge k \neq n$ holds and $a[k] \neq p$

and $s = 0$, however, we seem to have to switch to a different favorite candidate, i.e., to modify p. Preservation of $J0$ under modification of p requires an upperbound for the number of votes for the old value of p. It is the intention that the number of such votes increases with s and k. In this way, we come to the additional invariant

$$J1 : \quad 2 \cdot \mathit{votes}(k, p) \le s + k \ .$$

When $J1 \wedge k < n$ holds (and $a[k] = p$), then $J1$ is preserved when s and k are incremented. When $J1 \wedge k < n$ holds and $a[k] \ne p$ and $s > 0$, then $J1$ is preserved when k is incremented and s is decremented.

For the remaining case, we observe that $J0 \wedge J1 \wedge s = 0$ is equivalent to

$$s = 0 \quad \wedge \quad k \le n \quad \wedge \quad (\forall\, q :\ 2 \cdot \mathit{votes}(k, q) \le k) \ .$$

It is therefore preserved under arbitrary assignments to p, in particular under the assignment $p := a[k]$. In that way the troublesome remaining case can be avoided. We thus get the loop body

```
S :   if $s = 0$ then $p := a[k]$ end ;
        $\{J0 \ \wedge\ J1 \ \wedge\ B \ \wedge\ (s > 0 \ \vee\ a[k] = p)\}$
      if $p = a[k]$ then $s := s + 1$ else $s := s - 1$ end ;
      $k := k + 1$ .
```

The arguments given above show that S satisfies the Hoare triple

$$\{J0 \wedge J1 \wedge B\} \quad S \quad \{J0 \wedge J1\} \ .$$

The variant function $\mathit{vf} = n - k$ remains nonnegative because of $J0$. It clearly decreases under command S. This proves that postcondition Q is established by

```
$k := 0$ ; $s := 0$ ;
while $k \ne n$ do $S$ end .
```

Remark. If the program terminates with the final value $s = 0$, clearly no candidate holds a majority. If $s > \frac{1}{2} \cdot n$ in the postcondition, then p holds a majority. To prove this, however, we need the additional (but obvious) invariant $s \le \mathit{votes}(k, p)$.