The majority vote according to J. Gutknecht

I recently received from J. Gutknecht (ETH, Zürich) a nice solution to the problem known as "the majority vote", and one of the purposes of this note is just to record it. Its other purpose is to give a formal derivation of it, so that we can see the essence of Gutknecht's invention. Let me quote Gutknecht's statement of the problem:

"Let every inhabitant of a (non-empty) democracy be eligible as president. Let $b(i: 0 \le i < M)$ be a series of ballots. Develop a program that eliminates all but one candidate x, where no candidate eliminated has a majority of votes."

The formal statement of the postcondition to be satisfied by x is

R: $\quad (\underline{A} y: y \ne x: (\underline{N} i: 0 \le i \wedge i < M: b.i = y) * 2 \le M)$.

(Note that it is not required that x has a majority of votes: if none of the candidates has a majority of votes, any value for x satisfies R .)

* * *

An obvious candidate for the invariant is

P0: $\quad (\underline{A} y: y \ne x: (\underline{N} i: 0 \le i \wedge i < m: b.i = y) * 2 \le m)$

since it can be established by $m := 0$ and $[m = M \wedge P0 \Rightarrow R]$ (by construction of P0) .

What about its invariance under $m := m+1$?

$$
\begin{array}{ll}
 & wp.\text{"}m := m+1\text{"}.\,P0 \\
= & \{\text{axiom of assignment, definition of } P0\} \\
 & (\underline{A}y : y \neq x : (\underline{N}i : 0 \leq i \wedge i < m+1 : y = b.i) * 2 \leq m+1) \\
\Leftarrow & \{\text{properties of } \underline{N}\text{, definition of } P0\} \\
 & P0 \wedge (\underline{A}y : y \neq x : y \neq b.m) \\
= & \{\text{trading to } (\underline{A}y : y = b.m : y = x)\text{; one-point rule}\} \\
 & P0 \wedge x = b.m \quad ,
\end{array}
$$

which leaves the $x \neq b.m$ to be investigated.

Gutknecht's first invention is the introduction of a variable, s say, which records an upper bound on the number of "seen" votes for any currently eliminated candidate, i.e.

$$P1: \quad (\underline{A}y : y \neq x : (\underline{N}i : 0 \leq i \wedge i < m : y = b.i) \leq s) \quad .$$

By a calculation very similar to the above, we can establish

$$[P1 \wedge x = b.m \Rightarrow wp.\text{"}m := m+1\text{"}.\,P1]$$

Since currently eliminated candidates don't have a majority of the votes "seen", we can maintain —and this Gutknecht's second invention— $2 * s \leq m$ or

$$P2: \quad s \leq m - s \quad , \text{ established by } m, s := 0, 0 \ .$$

$P2$ is trivially invariant under $m := m+1$.

We can now forget about the invariance of $P0$ because $[P1 \wedge P2 \Rightarrow P0]$.

<u>Note</u> We could have derived $P2$ as the weakest solution of $P2: [P1 \wedge P2 \Rightarrow P0]$; then Gutknecht's second invention would have been to replace $P0$ by the conjunction of $P1 \wedge P2$. (End of Note.)

Now we return to the investigation how to increase m by 1 under invariance of $P1 \wedge P2$. in the case $x \neq b.m$. Because - for any B -

$$(\underline{N}i: 0 \leq i \wedge i < m+1: B.i) \leq (\underline{N}i: 0 \leq i \wedge i < m: B.i) + 1$$

$m, s := m+1, s+1$ maintains invariant $P1$. For the other conjunct of the invariant we investigate

$$
\begin{array}{ll}
 & wp. \text{"}m, s := m+1, s+1\text{"}. P2 \\
= & \{\text{axiom of assignment, definition of } P2\} \\
 & s+1 \leq m+1 - (s+1) \\
= & \{\text{arithmetic}\} \\
 & s < m - s
\end{array}
$$

So we can deal with the case $x \neq b.m \wedge s < m-s$; the only case left is $x \neq b.m \wedge s = m-s$. Here, Gutknecht remarked that there is no assignment to x yet, and his third invention is to consider for this case $m, x := m+1, b.m$. Since this assignment obviously maintains $P2$, we investigate the invariance of $P1$

$$
\begin{array}{ll}
 & wp. \text{"}m, x := m+1, b.m\text{"}. P1 \\
= & \{\text{axiom of assignment, definition of } P1\}
\end{array}
$$

$(\underline{A}y : y \neq b.m : (\underline{N}i : 0 \leq i \wedge i < m+1 : y = b.i) \leq s)$

$=$ {properties of $\underline{N}$}

$(\underline{A}y : y \neq b.m : (\underline{N}i : 0 \leq i \wedge i < m : y = b.i) \leq s)$

$\Leftarrow$ { because $x \neq b.m$, the second conjunct is needed; one-point rule}

$P1 \wedge (\underline{N}i : 0 \leq i \wedge i < m : x = b.i) \leq s$

$=$ { exploitation of $s = m - s$}

$P1 \wedge (\underline{N}i : 0 \leq i \wedge i < m : x = b.i) \leq m - s$.

And, finally, comes Gutknecht's optimism! Let us investigate whether we are lucky and P3, given by

$$P3 \qquad (\underline{N}i : 0 \leq i \wedge i < m : x = b.i) \leq m - s$$

is an invariant. It is established by the initialization $m, s := 0, 0$. We investigate our three cases.

$x = b.m \rightarrow m := m+1$

$wp.\text{"}m := m+1\text{"}.P3$

$=$ {axiom of assignment, definition of P3}

$(\underline{N}i : 0 \leq i \wedge i < m+1 : x = b.i) \leq m+1 - s$

$=$ { $x = b.m$ }

$(\underline{N}i : 0 \leq i \wedge i < m : x = b.i) + 1 \leq m+1 - s$

$=$ { arithmetic; definition of P3}

$P3$.

$x \neq b.m \wedge s < m-s \rightarrow m, s := m+1, s+1$

$wp.\text{"}m, s := m+1, s+1\text{"}.P3$

$=$ {axiom of assignment, definition of P3}

$(\underline{N}i : 0 \leq i \wedge i < m+1 : x = b.i) \leq (m+1) - (s+1)$

$$
\begin{array}{ll}
= & \{x \neq b.m \;;\; \text{arithmetic}\} \\
& (\underline{N}i{:}\ 0 \leq i \wedge i < m{:}\ \ x = b.i) \leq m - s \\
= & \{\text{definition of } P3\} \\
& P3 \qquad .
\end{array}
$$

$\underline{x \neq b.m \wedge s = m-s \rightarrow m, x := m+1, b.m}$

$$
\begin{array}{ll}
& wp.\text{"}m, x := m+1, b.m\text{"}, P3 \\
= & \{\text{axiom of assignment, definition of } P3\} \\
& (\underline{N}i{:}\ 0 \leq i \wedge i < m+1{:}\ b.m = b.i) \leq m+1-s \\
= & \{\text{properties of } \underline{N}\text{, arithmetic}\} \\
& (\underline{N}i{:}\ 0 \leq i \wedge i < m{:}\ b.m = b.i) \leq m-s \\
= & \{s = m-s\} \\
& (\underline{N}i{:}\ 0 \leq i \wedge i < m{:}\ b.m = b.i) \leq s \\
\Leftarrow & \{\text{instantiation with } y := b.m \;;\; b.m \neq x\} \\
& P1 \qquad .
\end{array}
$$

Thus the invariance of $P1 \wedge P2 \wedge P3$ has been established, and we have derived the program

```
|[ var m,s: int;  x, m, s := any, 0, 0
; do  m ≠ M →
       if  x = b.m  →  m:= m+1
       [] x ≠ b.m →
            if  s < m-s  → m, s := m+1, s+1
            [] s = m-s  → m, x := m+1, b.m
            fi
       fi
   od
]|                 ,
```

in which derivation I forgot -as usual!- to include

$0 \leq m \wedge m \leq M$ in the invariant; similarly, the proof of termination has been left to the reader.

* * *

The above derivation more than confirms my rule of thumb that the derivation of a non-trivial program is at least 10 times as long as the raw code in which it culminates; my formal manipulations and the identification of Gutknecht's inventions fully confirms that the majority vote algorithm — originally due to Boyer & Moore, be it in a different coding — is <u>not</u> trivial. So does the piece of luck that P_3 is invariant. (In his letter to me, Gutknecht adorned his program with 3 lines of problem statement and 5 lines of explanation, which by my standards, is a bit meagre. Hence this note.)

My indebtedness to Gutknecht is obvious.

Austin, Wednesday 25 January 1989

prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712 - 1188
United States of America