Shmuel Safra's version of termination detection

The other day, Shmuel Safra, a student of Amir Pnueli's, showed us his solution to a variation of the problem W.H.J. Feijen, A.J.M. van Gasteren and I solved in AvG25/WF52/EWD840, the variation being that message transmission no longer needs to be instantaneous.

We wrote that note "Derivation of a termination detection algorithm for distributed computations" not so much for the detection algorithm as for its derivation. This note is written to record how Safra's algorithm can be derived along the very same lines. In order to ease the comparison with EWD840 I shall use the same terminology. In order to keep this note self-contained I shall restate the problem.

* * *

We consider N machines, each of which is either active or passive. Only active machines send what are called "messages" to other machines; each message sent is received some finite period of time later. After having received a message, a machine is active; the receipt of a message is the only mechanism that triggers for a passive machine its transition to activity. For each machine, the transition from the active to the passive state may occur "spontaneously".

From the above it follows that the state in which all machines are passive and no messages are on their way is stable: the distributed computation with which the messages are associated is said to have terminated. The purpose of the algorithm to be designed is to enable one of the machines, machine nr.0 say, to detect that this stable state has been reached. For brevity's sake we shall denote the process by which termination is detected as "the probe".

The probe obviously has to involve, in some way or another, all machines. For simplicity's sake we adopt a circular arrangement, more precisely, we assume the availability of communication facilities such that

(i) machine nr.0 can initiate the probe by sending a signal to machine nr. N-1

(ii) machine nr. i+1 can propagate the probe around the ring by sending a signal to machine nr. i .

These signalling facilities are assumed to be available irrespective of the facilities for message transmission. Note that being passive (with respect to the distributed computation proper) does not prevent a machine from partaking in the above signalling.

The propagation of the probe around the ring allows us to describe that probe as a token being sent around the ring. The token being returned to machine nr.0 will be an essential component of the justification of the conclusion that the stable state has been reached.

As usual, the system state will be captured by an invariant, P say. In the sequel P will be constructed in a number of steps, each step consisting of an extension of the state space considered and an appropriate adjustment of P.

* * *

In the following, t denotes the index of the machine at which the token resides. The probe ends with $t=0$.

With B = (the number of messages on their way), termination is characterized by

$$(\underline{A}i: 0 \leq i < N: \text{machine nr.}i \text{ is passive}) \wedge B = 0$$

and, in order to detect termination, this has to be concluded from (i) the invariant, (ii) $t=0$, and (iii) further information available at machine nr.0. So far, information about B is available in none of the machines. This can be remedied by equipping each machine with its own counter c and so constructing it that it maintains

$P0$: $\quad B = (\underline{S}i: 0 \leq i < N: c.i)$.

The obligation to maintain $P0$ leads to

Rule 0 The sending of a message by machine nr.i includes $c.i := c.i + 1$; the receipt of a message by machine nr.i includes $c.i := c.i - 1$.

At the start of the computation proper we assume all c's properly initialized (e.g. with all c's $= 0$, corresponding to a start with empty channels).

Remark In view of the fact that B is in- and decremented distributedly, a distributed, additive representation of B, as captured in $P0$, is in essence our only option. (End of Remark.)

In order to enable the conclusion of termination upon return of the token, we strengthen the invariant to $P0 \wedge P1$ with $P1$ given by

$P1$: $(\underline{A}i: t < i < N:$ machine nr.i is passive$) \wedge$
$(\underline{S}i: t < i < N: c.i) = q$

where q is the value of the now integer token. It allows the conclusion of termination as

$$P0 \wedge P1 \wedge t = 0 \;\Rightarrow$$
$$(\underline{A}i: 0 < i < N:\text{ machine nr.i is passive}) \wedge$$
$$c.0 + q = B$$

so that upon return of the token, termination can be concluded from passivity of machine nr.0, in combination with $c.0 + q = 0$.

As $t,q := N-1, 0$ establishes P_1, we get

Rule 1 At probe initiation, machine nr.0 sends the token with $q=0$ to machine nr. $N-1$.

In order that token transmission, i.e. $t := t-1$, maintains P_1, we introduce

Rule 2 Machine nr. $i+1$ owning the token keeps it if active; if passive, it transmits it to machine nr.i under $q := q + c.(i+1)$.

Only message traffic involving machine nr.i with $t<i<N$ can - and does - falsify P_1. Since under P_1, such a machine is passive and does not send messages, falsification of P_1 takes place only under the initial truth of $B \geq 1$, i.e. - on account of P_0 - in terms of the variables introduced under P_2, given by

$$P_2: \quad (\underline{S}i: 0 \leq i \leq t: c.i) + q > 0 \quad ,$$

which remains true under message reception by machine nr. i with $t<i<N$.

So we are lead to consider the weaker invariant $P_0 \wedge (P_1 \vee P_2)$. First we verify that it still allows the conclusion of termination; it does since under $t=0$, P_2 equivales $c.0 + q > 0$, with the result that $c.0 + q = 0$ implies $\neg P_2$ and hence $P_0 \wedge P_1$. Secondly

we investigate the possible falsification of P_2. Thanks to Rule 2, token transmission maintains P_2; in view of Rule 0, however, message reception – by machine nr.i with $0 \leq i < t$ – may falsify P_2. Such message reception maintains the weaker $P_0 \wedge (P_1 \vee P_2 \vee P_3)$ with P_3 given by

P_3: $(\underline{E}i : 0 \leq i < t :$ machine nr.i is black$)$

under adoption of

Rule 3 On receipt of a message, the receiving machine turns black.

Since under $t=0$, P_3 equivales "machine nr.0 is black", machine nr.0 can still conclude termination if, in addition, it is white. Token traffic may violate P_3, viz. when a black machine transmits the token. Such transmission maintains the weaker $P_0 \wedge (P_1 \vee P_2 \vee P_3 \vee P_4)$ with P_4 given by

P_4 : the token is black

under adoption of

Rule 4 A black machine nr.i+1 transmitting the token to machine nr.i transmits the token blackened.

Since (the token is white) $\Rightarrow \neg P_4$, information available at machine nr.0 at token return can still suffice to conclude termination.

* * *

If, when the token has returned and machine nr.0 is passive, the token is black or machine nr.0 is black or $c.0 + q \neq 0$, the probe has been unsuccesful, in the sense that termination cannot be concluded. This is remedied by adopting

Rule 5 After an unsuccessful probe, machine nr.0 initiates a next probe.

Without the possibility of transitions from black to white, such a next probe is guaranteed to be as unsuccessful as its predecessor. Therefore our next task is to investigate where we can whiten without falsifying $P0 \wedge (P1 \vee P2 \vee P3 \vee P4)$.

As initiating the probe maintains $P0$ and establishes $P1$, which are both colour-indendent we can safely adopt

Rule 6 At initiation of the probe, machine nr.0 whitens itself and the token.

Since whitening a machine can falsify only $P3$, but does not do so when that machine's index exceeds i, we can safely adopt

Rule 7 Upon transmission of the token to machine nr.i, machine nr. $i+1$ whitens itself. (Note that, according to Rule 4, its original colour may have influenced the colour of the token.)

The above whitening protocols suffice: after terminations the c's are constant, their sum equals 0,

and no machine turns black anymore. A probe started in that state ends with $c.0 + q = 0$ and all machines white; the next probe ends with the token white as well.

* * *

There are presumably endless variations on this algorithm. If receipt of a message always leaves the recipient active, we could have replaced P_3 and Rule 3 by

P_3': $(\underline{E}i: 0 \leq i < t:$ machine nr.i is active or black$)$

Rule 3' Upon the transition from active to passive, a machine turns black.

But neither the algorithm nor its variations are the point of this note, which is about the derivation strategy, which worked again. Acknowledgements are due to Shmuel Safra and the members of the ATAC that were present.

15 January 1987

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