

A special sorting elephant.

This note deals with a generalization of the problem dealt with in EWD608, that described an elephant inspired by the problem of the Dutch National Flag. In this still exploratory stage, extending our collection of well-documented and carefully analysed elephants seems the most effective way of coming to grips with what would be required from the mathematics involved in the design of elephants. Hence this note.

We consider an elephant comprising (perhaps besides other mosquitoes) N mosquitoes, numbered from 0 through $N-1$. Each mosquito starts with a bag of coloured pebbles; there are N different colours, also numbered from 0 through $N-1$. Mosquitoes can send pebbles to and receive pebbles from other mosquitoes; the purpose of the whole elephant is that each mosquito collects from the system all pebbles of its "own" colour.

In order to make the problem difficult enough to be instructive, we have imposed upon the construction of our mosquitoes two (absolutely artificial) constraints:

- 1) the construction of each mosquito must be independent of the value of N
- 2) each mosquito is "partially colourblind", i.e. colour inspection of a pebble by a mosquito can only determine, whether the pebble has its own colour or not, but no mosquito can distinguish between different foreign colours.

We consider mosquitoes with two neighbours, locally called "L" and "R": L is the neighbour from which it receives pebbles, R is the neighbour to which it sends pebbles. (As in EWD608, we shall confine ourselves to channels between them that will be used for one-way traffic only.)

The initially proposed structure for a mosquito is as given on top of the next page. The body of the outer repetition we call "a turn". At the beginning of a turn the bag c contains the pebbles collected thus far, the bag b contains the pebbles to be transmitted in this turn to R . A turn consists of two nested blocks. At the outer block entry the input channel from L is established, at the inner block entry the output channel towards R is established. The primary purpose of the inner block is to empty bag b

```

begin b, c: bag of pebbles;
      b, c:= b0, c0 {this represents the initialization};
      do non empty(b) →
turn:   begin L: channel; d: bag of pebble; y: pebble; d:= emptybag;
        begin R: channel; x: pebble;
          do non empty(b) →
            x:= anyfrom(b); b:= b - x; R!(x);
            if L?(y) → d:= d + y
              || non L?(y) → skip
            fi
          od
        end {exit R channel};
        do L?(y) → d:= d + y od {L channel destroyed};
        c:= c + ownsfrom(d); b:= foreignsfrom(d)
      end
    od
end

```

by sending its pebbles one at a time to R, and to signal completion to R by destruction of the channel with R. The secondary purpose of the inner block is to carry its share of the load of the outer block that collects the incoming pebbles from L into bag d, initialized empty at the beginning of the turn. For each pebble sent from b to R the inner block can accept a new pebble from L. The guard "L?(y)" allows selection when a new pebble has arrived from L, the guard "non L?(y)" allows selection (of the action "skip") when the channel with L has been destroyed and, hence, no more incoming pebbles are to be expected during this turn.

The statement "do L?(y) → d:= d + y od" after the inner block caters for the case that in this turn more pebbles are received than have been sent; its termination depends on the destruction of the input channel from L. (About the convention of the input guard becoming false upon destruction of the channel by the partner at its other side, we are now less enthusiastic than when we first explored its consequences, but that is another story.)

The last line of the turn does the sorting: the own pebbles from d

are collected into c , the foreign pebbles from d are collected into b , ready for retransmission during the next turn.

* * *

It is here that I must reassure the reader who has a lurking suspicion that an elephant built from N such mosquitoes in a ring won't work: his suspicion is only too justified. We ask him, however, to postpone that worry, which is a global one, and to follow our analysis patiently, because this note is also the exploration of the practicality of a here consciously chosen methodology. When dealing with networks (such as elephants built from mosquitoes), this methodology tries to separate local and global concerns as rigorously as possible. First we analyse as completely as possible each node in isolation, thereby establishing its properties with respect to its immediate neighbourhood. The aim of these local considerations is to discover such properties (and formulate them in such a way!) that, when we switch to our global concerns, we have the material ready from which to build up an inductive argument proving the desired properties of the elephant as a whole.

* * *

Let us first analyse a single turn. What assumptions about its immediate environment have to be made for its successful execution? These assumptions are of two very different kinds: we have what we could call "the unidirectional assumptions", involving either L or R but not both, and the "bidirectional assumptions" involving both L and R . (At the beginning of a turn, our mosquito performs first an outer block entry, and then an inner block entry. Because at each of these entries a channel is declared, the outer block entry --for such are the rules of our game-- is synchronized with a matching block entry in L ; similarly, our mosquito's inner block entry is synchronized with a matching block entry in R . That in L and R these matching block entries can occur in the order as prescribed by the structure of our mosquito is an example of a bidirectional assumption.) Faithful to our habit of trying to separate our concerns, we shall try to deal with the uni- and bidirectional assumptions separately. Let us deal with the unidirectional assumptions first (this is suggested in spite of the fact that they seem easier than the bidirectional assumptions, and it is in general a wise practice to tackle the most difficult aspects of a problem first).

For the successful execution of a single turn we can confine our attention for the assumptions about R to the following annotated extract:

```

{non empty(b)}                                     (1)
begin R: channel; x: pebble;
  do non empty(b) → x:= anyfrom(b); b:= b - x; R!(x) od
end {exit R channel}

```

which is successfully matched by a text (that refers to our mosquito as "L")

```

begin L: channel; y: pebble;
  L?(y); do L?(y) → skip od
end

```

expressing that after channel creation at least one pebble will be sent, that the sender terminates autonomously --the bag b contains a finite number of pebbles to start with-- and that the receiver will correctly accept all pebbles sent --it will not ask for pebbles not coming and will stay "alive" to accept them all-- and will then duly terminate.

Because

```

if B → S || non B → skip fi; do B → S od

```

is semantically equivalent to

```

do B → S od

```

the assumptions of turn with respect to L are duly expressed by the extract:

```

begin L: channel; y: pebble; d: bag of pebble; d:= emptybag;
  do L?(y) → d:= d + y od
end

```

which is matched by (1), even without the initial assumption "non empty(b)": the execution of a turn is such that the outgoing traffic is nonempty and finite, of the incoming traffic our mosquito only requires that it is finite, empty would be permissible.

So much for the analysis of a single turn. If b_0 is empty, the mosquito makes no turns at all. Otherwise it will terminate after the turn which leaves an empty b ; but whether a turn leaves an empty b or not, depends as a result of the final " $b:=$ foreignfrom(d)" entirely on what it has

received during that turn. If we number the turns $0, 1, 2, \dots$, denote by b_j the pebbles sent during turn j , and denote by d_j the pebbles received in turn j , then the above assignment establishes the relation

$$b_{j+1}^i = \text{foreignsfrom}^i(d_j^i) \quad (2)$$

where we have added the mosquito number "i" as superscript because the function "foreignsfrom" is only locally defined.

The pebble gain in mosquito nr.i during turn j equals

$$d_j^i - b_j^i \quad ;$$

in order to ensure the conservation of pebbles we decide that each mosquito shall make the same number of turns and identify

$$b_j^i = d_j^{i+1} \quad (3)$$

(i.e. $(i+1) \bmod N$ to be very explicit). In other words, our N mosquitoes have been arranged cyclically. (Note that with (3) we have shifted from local to global considerations; the cyclic arrangement is most definitely not surprising, because with mosquitoes with one input channel and one output channel is it the only arrangement that provides a path from each mosquito to each other.)

In order to discover the condition for simultaneous termination to be imposed upon the b_0^i , I found working backwards the easiest. The number of turns is uniformly $= 0$ if b_0^i is empty for all i . It is uniformly $= 1$, if each b_0^i is not empty and only contains pebbles of colour $i+1$. For each pebble in b^i we can define $\text{dist}^i =$ the number of steps along the ring needed to reach its destination: for a pebble of colour m in b^i we have

$$\text{dist}^i(m) = (m-i-1) \bmod N + 1 \quad .$$

For each non-empty b^i we can define $\text{rank}^i(b^i)$ as the maximum number of steps a pebble in the bag needs to travel along the ring in order to reach its destination:

$$\begin{aligned} \text{rank}^i(b^i) &= \max(\text{dist}^i(m) \mid m \text{ in } b^i) \\ &= \max((m-i-1) \bmod N + 1 \mid m \text{ in } b^i) \quad ; \end{aligned} \quad (4)$$

for an empty b^i we define $\text{rank}^i(b^i) = 0$. It is tempting to suppose that all mosquitoes will terminate after k turns if and only if $\text{rank}^i(b_0^i) = k$.

This is indeed the case.

Because on account of (2) and (3)

$$b_{j+1}^i = \text{foreignsfrom}^i(d_j^i) = \text{foreignsfrom}^i(b_j^{i-1}) ,$$

and because for any nonempty bag b on account of (4)

$$\text{rank}^i(\text{foreignsfrom}^i(b)) = \text{rank}^{i-1}(b) - 1$$

we conclude, combining the two above relations:

$$\text{rank}^i(b_{j+1}^i) = \text{rank}^{i-1}(b_j^{i-1}) - 1 .$$

This establishes the invariance of

$$(\underline{A} \ i: 0 \leq i < N: \text{rank}^i(b_j^i) = k - j) .$$

The necessary and sufficient condition for termination after the same number of steps is therefore that $\text{rank}^i(b_0^i)$ is independent of i ; in particular, all mosquitoes will nicely terminate after N turns if and only if each b_0 contains at least one pebble of its own colour: all initial ranks are then $= N$.

If the bag c in the final state should only contain own pebbles, the necessary and sufficient condition is clearly that each c_0 is free from foreign pebbles.

* * *

Our analysis tells us that the sorting elephant will do its job provided we are entitled to make the identification (3)

$$b_j^i = d_j^{i+1}$$

i.e. are entitled to identify the pebbles sent by a mosquito during its turn _{j} with the pebbles received by its right-hand neighbour during the latter's turn _{j} . This question falls apart into two subquestions:

- 1) can we identify turn _{j} ^{i} with turn _{j} ^{$i+1$} ?
- 2) if yes, can the traffic take place?

We can find the answer to these questions by applying the theorem of EWD643 to the simple network formed by a ring. In that case the theorem reduces to the following one:

Consider a cyclic arrangement of mosquitoes, each of them communicating

in strict alternation with its two neighbours. Direct from each mosquito an arrow towards the first of its neighbours it will communicate with. The system is then free from deadlock if and only if initially the arrows don't point all (cyclicly) in the same direction.

For the identification of turn_j^i with turn_j^{i+1} we apply the above theorem while taking the establishment of the channel as the act of communication the theorem refers to. If we have only mosquitoes as described on top of EWD642 - 1, we would have a glorious deadlock because, initially, each arrow would point from a mosquito to its left-hand neighbour.

One way of circumventing this would be the insertion of at least one additional buffer mosquito in the ring that would establish its channels in the inverse order. For reasons of efficiency it is then attractive to insert N buffer mosquitoes, more precisely: to associate with each main mosquito a buffer mosquito to its left. When each buffer mosquito has the same sensitivity for colours as the main mosquito it is associated with, it can detect that it should terminate its business after a turn in which it has transmitted to its main mosquito only pebbles of their own colour.

Another way out would be to try to design an alternative main mosquito and to constrain the ring to contain mosquitoes of both types --for reasons of efficiency nearly equal numbers of both types-- .

Having established that all mosquitoes make the same number of turns, we focus our attention to the traffic corresponding to a single turn.

We now apply the theorem of EWD643 to the acts of communication transmitting a single pebble. As with main mosquitoes of type EWD642 - 1, each mosquito transmits in a turn to the right before it receives from the left, also the traffic would give rise to deadlock and one way out is to invert within the turn the communication order in the buffer mosquitoes. The other way out would be also in this respect two types of main mosquitoes.

Let us try to design the buffer mosquito first.

```

buffer mosquito:
begin frgn: bool; frgn:= true;
  do frgn → frgn:= false;
    begin R: channel;
      begin L: channel; y: pebble;
        do L?(y) → frgn:= frgn or foreign(y);
          R!(y)
        od {L channel destroyed}
      end
    end {exit R channel}
  od
end

```

Because each mosquito maintains its R-channel as long as it has something to send to the right and each mosquito only stops receiving from the left when the channel from the left has been destroyed, lemma 4 of EWD463 is applicable and we conclude that destruction of channels cannot cause the system to get stuck. It could in principle get stuck because channels fail to get destroyed: the buffer mosquito can only destroy its R-channel after its L-channel has been destroyed by its left-hand neighbour. The main mosquito of EWD642 - 1, however, destroys its R-channel independently of the destruction of its L-channel and, therefore, the construction with the N buffer mosquitoes is safe and sound.

* * *

Our next effort was to construct the elephant without buffer mosquitoes but from two types of main mosquitoes. Besides the main mosquito as described on EWD642 - 1 --which we now call "a type 1 main mosquito"-- we introduce type 2 main mosquitoes, as described on top of EWD642 - 8. The type 2 main mosquito shares two characteristics with the buffer mosquito given above: its R-channel is introduced at the outer block entry, while its L-channel is introduced at its inner block entry, and the $L?(y)$ command precedes the $R!(x)$ command. The type 2 main mosquito was designed in the hope that an elephant could be constructed from a ring containing both types of main mosquitoes. The elephant can, indeed, be constructed, but under circumstances, however, it won't function properly.


```

type 2 main mosquito:
begin b, c: bag of pebbles;
    b, c := b0, c0 {this represents the initialization};
    do non empty(b) →
turn:    begin R: channel; x: pebble; d: bag of pebble; d:= emptybag;
        begin L: channel; y: pebble;
            do L?(y) → d:= d + y;
                if non empty(b) →
                    x:= anyfrom(b); b:= b - x; R!(x)
                || empty(b) → skip
            fi
            od {L channel destroyed}
        end;
        do non empty(b) → x:= anyfrom(b); b:= b - x; R!(x) od;
        c:= c + ownsfrom(d); b:= foreignersfrom(d)
    end {exit R channel}
od
end

```

A two-mosquito elephant already provides the example of malfunctioning. Let b_0 of the type 1 main mosquito contain more pebbles than b_0 of the type 2 mosquito. As soon as the type 2 main mosquito has detected "empty(b)" in its innermost block the $R!(x)$ is skipped without destruction of its output channel. As a result the type 1 main mosquito gets stuck at the alternative construct of its innermost block and as a result the whole elephant gets stuck. The failure to construct an elephant from a mixture of type 1 and type 2 main mosquitoes was instructive: it is a strong indication that we haven't found the proper kind of mosquito yet!

One way out of the problems might be to place the blame on the absence of an explicit termination signal for the message stream, a signalling which is now coupled to a block exit. A more drastic solution departs from the idea of the purely sequential mosquito, and allows concurrency within individual mosquitoes. In the following design I have indicated two concurrent blocks S1 and S2 by parbegin S1 || S2 parend .

```

main mosquito with internal concurrency:
begin b, c: bag of pebbles;
    b, c := b0, c0 {this represents the initialization};
    do non empty(b) →
turn:   begin d: bag of pebbles; d:= emptybag;
        parbegin L: channel; y: pebble;
            do L?(y) → d:= d + y od
        ||
            R: channel; x: pebble;
            do non empty(b) → x:= anyfrom(b); b:= b - x; R!(x) od
        parend {empty(b), both channels destroyed and d filled};
        c:= c + ownsfrom(d); b:= foreignersfrom(d)
        end
    od
end

```

The introduction of the mosquito with internal concurrency is in this elephant such a powerful device that I feel hardly inclined to investigate the consequences of explicit signalling of termination of the message stream. Notice that with the introduction of this last mosquito we have simplified the elephant drastically in many ways:

- 1) for the proof of the absence of deadlock with respect to channel creations we don't need the theorem of EWD643 anymore
- 2) for the proof of the absence of deadlock with respect to the acts of communication we don't need the theorem of EWD643 anymore
- 3) the whole elephant consists of a ring of identical main mosquitoes
- 4) our new mosquito is simpler than the main mosquitoes considered before, in which we had to interleave the "receiving" and the "sending" cycles in a rather contorted manner.

* * *

The trouble with the signalling function of channel destruction caused by block exit is that I cannot make a buffer mosquito for a two-way channel that in addition can signal termination in either direction. Or is the two-way channel to blame? More experiments are necessary.

Plataanstraat 5
5671 AL NUENEN
The Netherlands

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow