On the correctness of a design by Alain J.Martin.

Alain J.Martin invented the following problem and its solution. (He had more solutions, but here we confine our attention to one of them.)

The problem. We consider a finite number of customer mosquitoes M, each with its dedicated service mosquito m. Each customer mosquito is synchronized with its service mosquito m via three different communication commands, denoted by m.p!, m.c?, and m.v!. (As the messages are empty, the separation in input commands and output commands is somewhat artificial.) The text of each customer mosquito is:

```
M: begin do true → noncritical section;
                   m.p!; m.c?;
                   critical section;
                   m.v!
   od
end
```

(Here "m.p!" can be viewed as the initiation of P(mutex), "m.c?" as the completion of P(mutex), and "m.v!" as V(mutex).)

The problem is to design the service mosquitoes m and their interconnection in such a way that at any moment in time at most one customer mosquito M is engaged in its critical section.

The solution. The service mosquitoes m are placed in a ring; each service mosquito refers to its anti-clockwise neighbour as "L", to its clockwise neighbour as "R". Each service mosquito m has a local boolean "pri" (short for "privilege") initialized at false, except for one exceptional service mosquito.

```
m: begin var pri: boolean; pri:= (m is the exceptional one);
      do M.p? → do non pri → R.p!; R.c?; pri:= true od; M.c!; M.v?
       [] L.p? → do non pri → R.p!; R.c?; pri:= true od; pri:= false; L.c!
      od
end
```

(The reader needn't be bothered by the redundant coding; we hope that the

redundancy will ease the discussion.)

* * *

The above design is worthy of our attention because in its structure it differs very much from the (ring-shaped!) designs we have considered before. I shall try to convince (myself and) my readers of its correctness by what seems the most effective --be it perhaps ad hoc-- argument. (Whether this argument, or parts of it, fit in a more general pattern that we can make respectable, is an important, but later concern.)

* * *

The mutual exclusion of the critical sections is easily proved. Because in M

m.c?; critical section; m.v!

coincides by definition with

M.c!; M.v?

in its service mosquito, it suffices to prove the mutual exclusion of the latter. Because during the latter the pri of the corresponding m is true, mutual exclusion is guaranteed if we can show that at most one pri-value is true.

By placing funny brackets < R.c?; pri:= true > (twice) and < pri:= false; L.c! > (once) , we see, because each R.c? coincides by definition with an L.c! , that at any moment --with this invention of atomic actions!-- there is exactly one pri-value true. The mutual exclusion is therefore guaranteed. (The above invention of the atomic actions seems a suitable candidate for being made more respectable!)

* * *

The above trick of combining communication commands into atomic actions --what about calling them "point actions" like the "point masses" in mechanics?-- nicely reflects that they embody mutual coincidence, and I shall try to apply it again. Let each customer mosquito M have an auxiliary boolean b0 , initialized at true , and let each service mosquito m have two auxiliary booleans, both initialized at false. For the customer mosquito I propose the following text:

```
M: begin var b0: boolean; b0:= true;
      do true → noncritical section;
                < m.p!; b0:= false > ; < m.c?; b0:= true > ;
                critical section;
                m.v!
      od
   end
```

For the service mosquitoes we depart more from the original text and also express syntactically that the inner repetitive constructs will be executed at most once:

```
m: begin var b1, b2, pri: boolean;
      b1, b2, pri := false, false, (m is the exceptional one);
      do < M.p? → b1:= true > ;
                 if pri → < b1, b2 := false, true >
                  [] non pri → < R.p!; b1:= false > ;
                               < R.c?; b2:= true; pri:= true >
                 fi;
                 < M.c!; b2:= false > ; M.v?
       [] < L.p? → b1:= true > ;
                 if pri → < b1, b2 := false, true >
                  [] non pri → < R.p!; b1:= false > ;
                               < R.c?; b2:= true; pri:= true >
                 fi;
                 < pri:= false; L.c!; b2:= false >
      od
   end
```

Because each query command --with the exception of M.v? -- sets a false boolean b0 , b1, or b2 to true, and each exclamation command --with the exception of m.v!-- sets a true boolean false, the number of true b's (b0, b1, or b2) remains constantly equal to its initial value, i.e. the number of (M,m)-pairs in the ring. For each service mosquito we further observe the invariant truths of

non (b1 and b2) and pri or non b2 .

When all customer mosquitoes are in their noncritical section, all b0's are true, hence all b1's and b2's are false, hence all service mosquitoes are ready to honour the query guards "M.p?" and "L.p?" . The latter, however, cannot compete with the "M.p?" , because no m-mosquito is ready for the execution of "R.p!" . Hence, if a number of M-mosquitoes has completed its noncritical section and is ready to execute "m.p!" , at least one will be able to do so and to set its b0 false.

Next we shall show how, when one or more b0's are false, a customer mosquito will be admitted to its critical section in a finite period of time; because each critical section is finite, it suffices to consider the situation in which none of the customer mosquitoes is engaged in its critical section.

For each b0 turned false, a b1 has been turned true in the first outer alternative --i.e. at the execution of < M.p? → b1:= true > -- of the corresponding m-mosquito. If this happened in the service mosquito with pri true , the true b1 wil be exchanged for a true b2 --by "b1, b2 := false, true"-- , the corresponding customer will then be admitted to its critical section. Otherwise a true b1 can only travel clockwise --via < R.p!; b1:= false > and < L.p? → b1:= true > -- ; the one with the minimum clockwise distance from it to the mosquito with pri true will do so until it has reached that service mosquito, where it will be exchanged for a true b2 --by < b1, b2 := false, true > in the second outer alternative-- . A true b2 can only travel anti-clockwise --via < pri:= false; L.c!; b2:= false > and < R.c?; b2:= true; pri:= true > -- ; on its anti-clockwise trip it takes the true pri as a companion with it, until it returns to the nearest mosquito waiting for it in the first outer alternative, where it will be exchanged --via < M.c!; b2:= false > and < m.c?; b0:= true > -- for a true b0 , and the customer mosquito is allowed to enter its critical section.

I don't regard the rather operational argument of the above paragraph as fully satisfactory. We could have made it more explicit by making "R.p!" and "L.p?" transmit as message the number of the m-mosquito that in-

troduced --via < M.p? → b1:= true > -- the true b1 into the m-ring . (I shall not do so, the reader can do so for himself, if he so desires.) The argument that true b1's can only travel clockwise, but cannot do so indefinitely, because one will hit the mosquito with pri true, in which it will be reflected as a true b2 , which can only travel anti-clockwise, but cannot do so indefinitely because it will be absorbed by the m-mosquito that introduced into the m-ring the true b1 from which it ultimately originated, is only ok in the absence of deadlock: we have argued that we have described the only possible history, but will it happen? The next section of this note will therefore be devoted to deadlock analysis.

* * *

From the text of the M-mosquitoes we derive syntacticly that its communication pattern is given by

$$\{m.p!\ ,\ m.c?,\ m.v!\} \quad . \tag{1}$$

From the text of the m-mosquitoes we derive similarly the syntax for the communication pattern

$$\{M.p?,\ (\ |\ R.p!,\ R.c?),\ M.c!,\ M.v?\ |\ L.p?\ ,\ (\ |\ R.p!,\ R.c?),\ L.c!\} \tag{2}$$

When we "project" this grammar on M --i.e. omit all communications not communicating with M -- we get

$$\{M.p?,\ M.c!,\ M.v?\} \tag{3}$$

and observe that (1) and (3) match beautifully. Projecting (2) on R we get

$$\{\ (\ |\ R.p!,\ R.c?)\ |\ (\ |\ R.p!,\ R.c?)\ \}$$

which reduces to

$$\{R.p!,\ R.c?\} \quad ; \tag{4}$$

projecting (2) on L , we get by a similar reduction

$$\{L.p?,\ L.c!\} \tag{5}$$

and observe that also (4) and (5) match beautifully with each other.

Due to the absence of alternatives as would be indicated by "|" and of nested brace pairs, the matching syntaxes can generate each only one infinite sentence. As a result the possibility of a "local conflict" is

excluded, where with a local conflict we mean the situation in which the partners at the two ends of a channel are ready to communicate with each other, but with nonmatching commands, i.e. commands with different labels or without query/exclamation matching. In passing we conclude that in this example the labelling with "p", "c", and "v" is superfluous.

An earlier version of this text proceeded at this point with the proof of the absence of the danger of deadlock, because without further assumptions the absence of the danger of individual starvation cannot be proved. In retrospect it is more efficient to make the further assumption and to proceed directly to a proof of the --in general stronger-- statement that the danger of individual starvation is absent. The further assumption is that the nondeterminacy in the repetitive construct of each service mosquito is resolved by a fair daemon, more precisely

1) that a customer mosquito permanently blocked at its "m.p!" implies that the nondeterminacy of its service mosquito has only been resolved a finite number of times; with the assumption of finite speeds this implies that its service mosquito is eventually permanently blocked at one of its communication commands inside the first or the second outer alternative

2) that a service mosquito permanently blocked at its "R.p!" implies that the nondeterminacy of its clockwise neighbour has only been resolved a finite number of times; with the assumption of finite speeds this implies that its clockwise neighbour is eventually permanently blocked at one of its communication commands inside the first or the second outer alternative.

On account of the matching between (1) and (3), the grammar (2) tells us that no service mosquito can be permanently blocked on its "M.c!" or its "M.v?" . (This conclusion remains valid if the requirement of finite speed is relaxed for a customer mosquito engaged in its noncritical section.)

Furthermore syntax (2) tells us that each "R.c?" has been preceded by its "R.p!", which, if "R.c?" causes blocking, has been honoured by the "L.p?" of its clockwise neighbour. Again syntax (2) tells us that this blocking --i.e. the inability of the clockwise neighbour to perform the matching "L.c!"-- implies that the clockwise neighbour is blocked at "R.p!" or "R.c?" .

Finally syntax (2) tells us that no service mosquito can ever be blocked permanently on its "L.c!" , as it has been preceded by its "L.p?" which has been honoured by the "R.p!" of its anti-clockwise neighbour, which by doing so becomes ready to honour via "R.c?" the "L.c!" in question.

As a result, a customer mosquito permanently blocked at "m.p!" implies its service mosquito permanently blocked at either "R.p!" or "R.c?"; and a service mosquito permanently blocked at either "R.p!" or "R.c?" implies that its clockwise neighbour is permanently blocked at either "R.p!" or "R.c?". Via induction we conclude that a customer mosquito permanently blocked at "m.p!" implies that all service mosquitoes are blocked at either "R.p!" or "R.c?". As this implies that all pri-values are false, we conclude that no customer mosquito will be permanently blocked at "m.p!".

A customer mosquito permanently blocked at "m.c?" --to be matched by "M.c!" of its service mosquito-- would also imply that this service mosquito is permanently blocked by either "R.p!" or "R.c?"; the same argument allows us to conclude that no customer mosquito will be permanently blocked by its "m.c?".

That no customer mosquito will be permanently blocked at "m.v!" is obvious from grammars (1) and (2).

With the above the absence of the danger of deadlock has been demonstrated. We note --for later?-- that in this last argument the auxiliary variables b0 , b1 , and b2 have played no role whatsoever: the grammars (1) through (5) --of which only (1) and (2) are independent-- gave all the information, together with the knowledge of one true pri-value in the ring.

Plataanstraat 5
5671 AL NUENEN
The Netherlands

prof.dr.Edsger W.Dijkstra
BURROUGHS Research Fellow