Copyright Notice

The manuscript was published as pages 259–263 of

Edsger W. Dijkstra, *Selected Writings on Computing: A Personal Perspective*, Springer-Verlag, 1982. ISBN 0–387–90652–5.

A correctness proof for communicating processes: a small exercise.

Over the last one-and-a-half year C.A.R.Hoare has explored "communicating sequential processes", among many other targets as a means for describing "elephants built from mosquitoes, all humming in harmony", to quote the old metaphor. His approach has two main characteristics to be described now.

1) The so-called "marriage bureau coupling". Inspired by our familiarity with the assignment statement, he has decided to try to visualize in- and output as the two sides of an assignment statement. In the one mosquito the input command assigns a value to one of its --by definition!-- private variables, in the other mosquito the matching output command provides the value to be assigned. In the implementation these in- and output commands are supposed to prescribe an implicit synchronization: they are viewed as completed simultaneously. (This is in accordance with our earlier impression, viz. that "mutual coincidence" is in such an environment a more essential notion than "mutual exclusion".)
Given:

mosquito "x" with a local variable "a"

mosquito "y" with a locally formed value "E"

then the "simultaneous" execution of their respective commands:

y?(a) x!(E)

is semantically equivalent to

a:= E .

Note that the program text for mosquito "x" mentions the sender "y" in its input command "y?(a)", and that in the text for mosquito "y" the receiver "x" is mentioned in its output command "x!(E)" .

2) Each pair of mosquitoes is connected via at most a single channel that accommodates two-way traffic. This imposes an ordering in time of the acts of communication between any two mosquitoes. It was felt that this would simplify the mathematical treatment.

* * *

We embarked upon one of a series of examples of communicating sequential processes solving a sorting problem suggested by Wim H.J.Feijen. Two mosquitoes each start with a "bag of natural numbers" --the difference between a "bag" and a "set" being that in a bag not all elements need to be different from each

other-- . Mosquito x removes the maximum value from its bag and sends it to mosquito y , which adds it to its bag; this is followed by a transmission by y to x of the minimum element taken from the bag of y , etc. Eventually x ends up with the small elements in its bag and y with the large ones.

Our aim was to investigate to what extent the two mosquitoes could be successfully investigated in isolation. We wrote down texts for both mosquitoes, and then covered the one text with a piece of paper. I now simulate that by first only giving you the text for mosquito x (with many notational liberties of which I hope that they won't confuse you; $\simeq$ and $\not\simeq$ stand for addition to and removal from bags).

Mosquito x:

```
begin  r, s: bag of nat; a, p: nat;
       s:= S {the constant S is a non-empty bag of nat};  p:= max(s);
       y!(p); r:= s ≃ p;
       y?(a); s:= r ≄ a;
       p:= max(s) {P};
       do p > a → y!(p); r:= s ≃ p;
                  y?(a); s:= r ≄ a;
                  p:= max(s) {P}
       od
end
```

With sum(bag) = the sum of the numbers contained in "bag", we have as the relevant invariant relation for the do...od:

P = (sum(s) = sum(r) + a) and p = max(s) $\geq$ a .

The first equality is established after s:= r ≄ a , the inequality p $\geq$ a is established by p:= max(s) , because max(s) $\geq$ any element in s and the element "a" is in s.

We choose for the variant function sum(r):

wdec("r:= s ≃ p", sum(r)) = sum(s ≃ p) < sum(r) =
sum(s) - p < sum(r) = {on account of P}
sum(r) + a - p < sum(r) = p > a .

Hence the guard "p > a" guarantees effective decrease of sum(r) .

Because natural numbers are bounded from below, ~~sum(r) is bounded from below~~, sum(r) is so too, and mosquito x terminates. In its final state it has established (P and $p \leq a$), which implies $max(s) = a$, i.e. the final value of "a" occurs in the bag "s" and is the largest value in that bag.

(If the value(s) of "a" would not be bounded from below, termination, indeed, cannot be guaranteed. I shall not pursue that now, because proofs of nontermination are a different story.)

* * *

We now turn our attention to mosquito y.

```
begin t, u: bag of nat; b, q: nat;
      t:= T {the constant T is a nonempty bag of nat};
      x?(b); u:= t ∓ b;
      q:= min(u);
      x!(q); t:= u ≃ q {Q};
      do x?(b) → u:= t ∓ b;
                 q:= min(u);
                 x!(q); t:= u ≃ q {Q}
      od
end
```

The "query guard" x?(b) is regarded to have the side-effect of assigning a value to b when evaluating to true --as a matter of fact, the value transmitted by the matching y!(p) in mosquito x, but the discussion of this interaction is postponed, as well as the discussion of how a happening in mosquito x can cause the query guard x?(b) to become false--. The invariant relation Q for y's repetitive construct that interests us is

$$Q = \quad q \leq min(t)$$

We have $wp("t:= u \simeq q", Q) = q \leq min(u \simeq q)$; because $min(u \simeq q) \geq min(u)$, the previous weakest pre-condition is implied by $q = min(u)$, a relation which is established by $q:= min(u)$. In short: when mosquito y has terminated, it has established $q \leq min(t)$, i.e. all ~~elements~~ elements in the bag t are greater than or equal to the final value of q (the final value of q need not occur in the bag t).

* * *

The proofs, so far, have surprised us in two respects. First of all: when we started we did not know that the weakest condition on the input stream of the a's for termination of x would be that the a's are bounded from below and nothing else . (I believe that intuitively I felt, that the sequence of a's being non-increasing had something to do with it; quod non.) Secondly, we feared another complication when we started: mosquito x terminates when otherwise it would send a value p = to the value "a" just received. This value has been transmitted once --if originally in T -- or twice --if originally in S -- , and for that reason we expected that we would have to distinguish between those two cases. (Trying to live with sum(s) as variant function would have introduced similar problems.) In our treatment the distinction between those two cases has disappeared completely --I even hope that some of my readers did not realize this distinction before I pointed it out to them!-- , and that is probably the most pleasant and encouraging gain that we derived from dealing with our mosquitoes in isolation. By now we have studied them to such an extent in isolation, that time has come to study the combination.

There are a few rules of the game: input/output command sequences at both sides of a channel must match, i.e. for an input command at the one side of the channel we must have a matching output command at the other side. Well, in this simple example, this is OK, in the sense that the sequence of channel commands in x are given by the syntax --with {...} denoting zero or more instances of the enclosed--

y!(p) y?(a) {y!(p) y?(a)}

and in mosquito y by

x?(b) x!(q) {x?(b) x!(q)} .

Ignoring the arguments p , a , b , and q , the one syntax can be transformed into the other by interchanging x and y and also interchanging ? and ! . Hence, both syntaxes contain matching sentences, and the whole thing will match, provided that from both syntaxes "the same" sentence is chosen. In this case the choice of sentence is restricted to the length: both mosquitoes must terminate at the same stage.

It seems very tricky if separate termination proofs for both mosquitoes must be given, with in addition a proof that they will terminate after the same amount of traffic. (Not impossible, but tricky.) One of the rules of the game

is that when one of the mosquitoes decides on account of its internal logic --such as x in this example-- to quit, that this can result in "disappearance of the channel" --e.g. by a block exit, not indicated in the text on pg-EWD607 - 1-- and that disappearance of the channel will cause at the other side communication commands in a guard position --such as the (second) x?(b) in the text for y -- to give rise to a false guard. Tony seems to have chosen for an asymmetry here: only "query guards" are allowed in his proposal. Although the decision is defensible, for the time being we would also like to allow"exclamation guards": termination because the receiving end decides that it has had enough! (Sorry for the very operational terminology.) In view of the symmetry between in- and output, this greater freedom does not seem to create much complication. With such an implicit convention for termination, the communication sequences at both ends are now forced to match. (The match can even be decided on purely syntactic grounds; we hope that this will always be the case.)

Associating with y!(p) the implicit assignment pp:= pp ⨾ p (on the "ghost bag" pp, which is initialized empty), and similarly with a "ghost bag" aa, associating with y?(a) thereafter aa:= aa ⨾ a , we can strengthen P with

$$s = S \Leftarrow aa \simeq pp \quad ;$$

similarly, Q can be strengthened with the relation

$$t = T \Leftarrow bb \simeq qq \quad .$$

Taking the arguments in our matching syntaxes into account, a postulate about the communication must enable us to identify p with b , hence pp with bb and a with q, hence aa with qq . And thus we find firstly $s \Leftarrow t = S \Leftarrow T$ i.e. conservation of elements. But it also allows us to equate the final value of "a" with the final value of "q" , this combining from the two final states

$$max(s) = a = q \leq min(t) \quad ;$$

thus the correctness of the elephant has been established.

Acknowledgements are due to all the countrymen (women) with whom I regularly talk about my work: Feijen, Rem, Scholten, Bulterman, Steffens, Martin etc. They are not to be held responsible for my mistakes or what have you.

Plataanstraat 5
NL- 4565 NUENEN
The Netherlands

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow