

A small note on the additive composition of variant functions.

This is only a small note, for its observation is nearly trivial. The note is made, however, because it describes a method of some generality, by which results can be derived that --as I have observed-- tend to strike the reader as "a trick" if he has never seen the like and is unaware of the method behind it.

It deals with finding a variant function for a repetitive construct with more than one alternative:

$$\underline{\text{do}} \ B1 \rightarrow S1 \ \parallel \ \dots \ \parallel \ Bn \rightarrow Sn \ \underline{\text{od}} \ .$$

The simple idea is to construct the variant function  $t$  as the sum

$$t = t1 + \dots + tn \ ,$$

where  $t_i$  is decreased by at least one by the  $i$ -th guarded command and left constant by the others.

We shall illustrate it with an example of matching simplicity. On a shunting yard are a finite number of trains; a train is either atomic (iff it consists of a single car) or composite (iff it consists of a finite number of cars greater than one). The yard is empty, when it contains no trains. Repeatedly we select a train: if it is atomic, it is removed from the yard, if it is composite, it is split into two (shorter) trains that, in this move, remain on the yard. The process terminates, when the yard is empty. The point is to construct the decreasing function  $t$  that is bounded from below, so that termination is proved. We sketch the process by

$$\begin{array}{l} \underline{\text{do}} \ \text{atomic train selected} \rightarrow \text{remove it} \\ \quad \parallel \ \text{composite train selected} \rightarrow \text{split it} \\ \underline{\text{od}} \end{array}$$

With  $\text{CARS} =$  the number of cars on the yard

$$t1 = \text{CARS}$$

satisfies the requirement that the first guarded command decreases it and the other one leaves it constant. With

$\text{TRAINS} =$  the number of trains on the yard ,

$$t2 = - \text{TRAINS}$$

would satisfy the requirement that it is decreased by the second guarded command

(because that increases + TRAINS). This remains true when we add a function left constant by it, e.g.

$$t2 = f(\text{CARS}) - \text{TRAINS} .$$

(Note that we have already verified that CARS, and therefore  $f(\text{CARS})$  for any function  $f$ , is unaffected by the second guarded command.) Because  $t2$  has to remain constant when the first guarded command is selected, we must choose for  $f$  the identity function:

$$t2 = \text{CARS} - \text{TRAINS}$$

and thus we find  $t = 2 * \text{CARS} - \text{TRAINS} .$

(With CLOSCONNS = the number of closed car-to-car connections,

$$t2 = \text{CLOSCONNS}$$

also does the job: it is left unchanged by the first and decreased by the second guarded command. This is not a new solution, for  $\text{CLOSCONNS} = \text{CARS} - \text{TRAINS} .$ )

\* \* \*

For the asymmetric computation for  $x = \text{gcd}(X, Y)$  with  $X > 0$  and  $Y > 0$  :

$x, y := X, Y$ ; do  $x > y \rightarrow x := x - y$   $\square$   $y > x \rightarrow x, y := y, x$  od

we find quite analogously  $t1 = x + y$  and  $t2 = y .$

\* \* \*

Note. Obviously the condition that  $t_i$  is left unchanged by the  $j$ -th ( $j \neq i$ ) guarded commands is stronger than necessary: it suffices to establish that  $t_i$  is not increased by the other guarded commands. (End of note.)

Note. Instead of choosing  $t = t1 + \dots + tn$ , we could also take --after having chosen the  $t_i$  --  $t = a1*t1 + \dots + an*tn$  with all  $a_i$ 's positive; in other words, the variant function is by no means unique, and we can choose the most convenient one. (End of note.)

The above has been prompted by my impression that some people seem to think that that there is something mysterious about the termination of the type of programs discussed. Its moral is that there is no mystery.

Plataanstraat 5  
 NUENEN NL-4565  
 The Netherlands

prof.dr.Edsger W.Dijkstra  
 Burroughs Research Fellow