A conversion routine revisited

In the following, $c(k: 1 \le k)$ is a sequence of integers satisfying

(0) $\qquad c_k \ge 2 \quad$ for all $k \ge 1$,

but, for the time being, only $c_k \ne 0$ is relevant: it allows us to define $w(k: \underline{0} \le k)$ by

(1) $\qquad w_0 = 1$

(2) $\qquad \underline{w_{k-1}} = c_k * w_k \quad$ for all $k \ge 1$.

We now consider the following program defining the sequence $d(k: 1 \le k)$ in terms of the real number R

```
    n,f := 0,R  {P0}
  ; do true → n:= n+1  {P1}
            ; d_n, f := INTFRA.(f* c_n)  {P0}
    od
```

where, for the time being, our only knowledge about the pair-valued function INTFRA of a real argument is

(3) $\qquad x, y = \text{INTFRA}.z \;\Rightarrow\; x + y = z$.

The above suffices to demonstrate the invariance of

P0: $\qquad R = (\Sigma k: 0 < k \wedge \underline{k \le n}: d_k * w_k) + f * w_n$

(leaving the fact that n is natural understood).

In order to verify that the initialization establishes $P0$ we observe

$$
\begin{aligned}
& wp.\text{"}n,f := 0,R\text{"}.\ P0 \\
= \quad & \{\text{axiom of assignment, definition of } P0\} \\
& R = (\Sigma k: 0<k \wedge k \leq 0: d_k * w_k) + R * w_0 \\
= \quad & \{\text{definition of empty sum; } (1)\} \\
& R = 0 + R * 1 \\
= \quad & \{\text{arithmetic}\} \\
& true \quad .
\end{aligned}
$$

In order to verify the invariance of $P0$ we observe —after introducing the abbreviation S:

$$S = \quad d_n, f := INTFRA.(f * c_n) \quad -$$

for $n \geq 1$

$$
\begin{aligned}
& wp.S.P0 \\
= \quad & \{\text{definition of } P0\} \\
& wp.S.(R = (\Sigma k: 0<k \wedge k \leq n: d_k * w_k) + f * w_n) \\
= \quad & \{\text{splitting the range, as } n \geq 1\} \\
& wp.S.(R = (\Sigma k: 0<k \wedge k \leq n-1: d_k * w_k) + \\
& \qquad (d_n + f) * w_n \ ) \\
= \quad & \{\text{axiom of assignment, definition of } S,\ (3)\} \\
& (\Sigma k: 0<k \wedge k \leq n-1: d_k * w_k) + f * c_n * w_n \\
= \quad & \{(2) \text{ with } k := n\} \\
& (\Sigma k: 0<k \wedge k \leq n-1: d_k * w_k) + f * w_{n-1} \\
= \quad & \{\text{definition of } P0 \text{ and of substitution}\} \\
& (n := n-1).\ P0 \\
= \quad & \{\text{our nomenclature}\} \\
& P1 \quad .
\end{aligned}
$$

Since [wp."n:=n+1".P1 ≡ P0], the invariance of P0 has now been established.

* * *

Our next target is

(4) $R = (\Sigma k : 0<k : d_k * w_k)$,

a relation which follows from P0 if

(5) $\lim_{k \to \infty} f * w_k = 0$.

In view of (0) —which now becomes relevant— and (2), $\lim_{k \to \infty} w_k = 0$, so (5) follows if f is bounded.

A look at the program tells us that boundedness of f depends on the function INTFRA, about which, up till now, only (3) has been given. For given z, the relation $x + y = z$ of (3)'s consequent can be satisfied in many ways: the one extreme, viz. $x=0 \wedge y=z$, leads to unbounded f, the other extreme, viz. $x=z \wedge y=0$ leads, instead of to the general (4), to $R = d_1 * w_1$, which is not interesting. So we choose something in between.

The left element of INTFRA.z is constrained by

(6) $x,y = $ INTFRA.z $\Rightarrow$ x is integer ,

which allows us to restrict y to a unit interval. For the constraint on y we consider the following two alternatives:

(i) $x,y = \mathrm{INTFRA}.z \;\Rightarrow\; 0 \le y \wedge y < 1$ and

(ii) $x,y = \mathrm{INTFRA}.z \;\Rightarrow\; 0 < y \wedge y \le 1$.

We now restrict our investigation of these alternatives to $0 < R \wedge R < 1$; in the following it becomes relevant that c is a sequence of integers.

Alternative (i) leads to the invariance of

$$0 \le f \wedge f < 1 \qquad ;$$

alternative (ii) leads to the invariance of

$$0 < f \wedge f \le 1 \qquad .$$

In both cases we have for all k

$$0 \le d_k \wedge d_k < c_k \qquad .$$

The minimum value 0 and the maximum value $c_k - 1$ can in both cases occur for d_k .

For a given sequence c , we call an R-value <u>sensitive</u> if its corresponding sequence d depends on whether (i) or (ii) has been chosen, otherwise we call it <u>insensitive</u>. We shall see below that for any sequence c , values of both types exist.

Because of $0<R \wedge R<1$, the initialization of the program establishes

(7) $\qquad 0<f \wedge f<1$.

If with one of the alternatives for INTFRA , $f*c_n$ is never integer, (7) is an invariant and the computation is indistinguishable from the one with the same R but the other alternative for INTFRA : R is insensitive. (E.g. $R=1/2$ for $c_k=3$.)

We now investigate the case in which, sooner or later, the repeatable statement falsifies (7). That occurs when, with f satisfying (7), $f*c_n$ is integer; let then

$$f*c_n, n = q, N$$.

Because f satisfies (7), q satisfies

$$1 \leq q \wedge q < c_N$$.

The rest of the computation depends on which alternative has been chosen for INTFRA. With (i), the assignment is effectively

$$d_N, f := q, 0$$,

and, $f=0$ being stable with (i), the program defines

$$(\underline{A}k: N<k: d_k = 0)$$,

i.e. sequence d ends with $d_N > 0$, followed by "the minimum tail".

With (ii), the assignment is effectively

$$d_N , f := q-1 , 1 \quad ,$$

and, $f=1$ being stable with (ii), the program defines

$$(\underline{A} k : N < k : d_k = c_k - 1) \quad ,$$

i.e. sequence d ends with $d_N < c_N - 1$, followed by "the maximum tail".

We conclude that an R giving rise to an integer $f * c_n$ is sensitive.

<u>Remark</u> With, for some positive N, $R = w_N$, R is sensitive. In view of (4), the above analysis tells us

(8) $\qquad w_N = (\Sigma k : N < k : (c_k - 1) * w_k) \quad .$

(End of Remark.)

If R and d satisfy (4), we say that "d is a representation of R", and in that terminology we just found that insensitive values have at least 1 representation and sensitive values have at least 2 representations. Under the constraint

$$(\underline{A} k : 0 < k : 0 \leq d_k \wedge d_k < c_k)$$

values don't have more representations. The argument relies primarily on the monotonicity of addition. We ask ourselves under what circumstances two different sequences can represent the same value. Let X be the common prefix of the two sequences, in the one followed by "p tail0" and in the other by "q tail1", with $p<q$. We now consider the following sequences and values

X p tail0	R0
X p maximum tail	R1
X p+1 minimum tail	R2
X q minimum tail	R3
X q tail1	R4

Because in tail0 $d_k < c_k$, $R0 \leq R1$ and $R0 = R1 \Rightarrow$ tail0 = maximum tail. Because of (8), $R1 = R2$. Because of $p<q$, $R2 \leq R3$; $R2 = R3 \Rightarrow p+1 = q$. Because in tail 1 $0 \leq d_k$, $R3 \leq R4$ and $R3 = R4 \Rightarrow$ minimum tail = tail 1. Because the sequence of R-values is ascending, $R0 = R4$ implies that they are all equal, hence

tail0 = maximum tail

$p+1 = q$

tail1 = minimum tail,

and the two sequences representing the sensitive value are the ones generated according to (ii) and (i) respectively. Also: a sequence not ending in an extreme tail is insensitive.

In other words, the shape of the sequence — i.e. whether or not it has an extreme tail — determines whether the value represented by the sequence is sensitive or not.

Each sensitive R is rational. The fraction $1/q$ is sensitive provided, for some N, q divides the product of the first N c-values. Hence, each rational R is sensitive provided

(9) $(\underline{A}q : 0<q : (\underline{E}N :: q \text{ divides } (\Pi k : 0<k \wedge k \le N : c_k)))$.

Combining the two observations, we conclude

$$(9) \Rightarrow (R \text{ is sensitive}) \equiv (R \text{ is rational}) \quad .$$

Condition (9) is clearly satisfied for $c_k = k+1$. Hence e , the basis of the natural logarithm, is irrational.

Nuenen, 6 January 1991

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