Fibonacci and the greatest common divisor

Let the function f (from naturals to naturals) be given by

(0) $f.0 = 0$, $f.1 = 1$, $f.(n+2) = f.(n+1) + f.n$.

Then, f application distributes over gcd , i.e.

(1) $f.(X \text{ gcd } Y) = f.X \text{ gcd } f.Y$.

* * *

Our interest is not in the above theorem, nor in its proofs. We wish to explore how we could design a proof for it.

We know the gcd for positive operands as the outcome of Euclid's Algorithm:

(2) $x, y := X, Y$
  ; do $x > y \rightarrow x := x - y$
   ▯ $y > x \rightarrow y := y - x$
  od $\{x = X \text{ gcd } Y \wedge y = X \text{ gcd } Y\}$,

and this knowledge raises the question of whether we can prove (1) by a properly chosen invariant for program (2) . Which invariant, true before the repeatable statement, allows us to conclude (1) upon termination? We observe, starting with the left-hand side of (1)

$$\begin{array}{ll} & f.(X \,\underline{\text{gcd}}\, Y) \\ = & \{ \underline{\text{gcd}} \text{ is idempotent: } Z \,\underline{\text{gcd}}\, Z = Z \} \\ & f.(X \,\underline{\text{gcd}}\, Y) \,\underline{\text{gcd}}\, f.(X \,\underline{\text{gcd}}\, Y) \\ = & \{ x = X \,\underline{\text{gcd}}\, Y \wedge y = X \,\underline{\text{gcd}}\, Y \} \\ & f.x \,\underline{\text{gcd}}\, f.y \\ = & \{ (3)\text{, see below} \} \\ & f.X \,\underline{\text{gcd}}\, f.Y \end{array}$$

with the suggested invariant (3) given by

(3) $\quad f.x \,\underline{\text{gcd}}\, f.y = f.X \,\underline{\text{gcd}}\, f.Y$.

Since (3) is obviously established by the initialization of (2), we only need to show that (3) is maintained by (2)'s repeatable statement, i.e. we have to show

$$f.(x-y) \,\underline{\text{gcd}}\, f.y = f.x \,\underline{\text{gcd}}\, f.y \quad \text{for } x>y \wedge y>0$$

or, more symmetrically written:

(4) $\quad f.a \,\underline{\text{gcd}}\, f.b = f.(a+b) \,\underline{\text{gcd}}\, f.b \quad$ for $a>0, b>0$.

It is obviously time to take into account what has been given about f .

The first step is to rewrite (0) a little bit more elegantly as

(5a) $\quad (f.0, f.1) = 0, 1$

(5b) $\quad (f.(n+1), f.(n+2)) = f.(n+1), f.(n+1) + f.n$.

In terms of pairs of successive f-values, i.e. $p.n = (f.n, f.(n+1))$, these equations have the form

(6) $\quad p.0 = (0,1) \quad , \quad p.(n+1) = F.(p.n) \qquad ;$

the advantage of (6) is that the introduction of the function F —from pairs of naturals to pairs of naturals— enables us to write the solution in closed form:

$$p.n = F^n.(0,1) \quad .$$

In view of our remaining proof obligation (4), we shall now try to exploit the associativity of function composition, in particular

(7) $$F^{a+b} = F^a \circ F^b \quad .$$

This exploitation requires that the specific shape of F is taken into account. Writing p.n as column vector

$$p.n = \begin{matrix} f.n \\ f.(n+1) \end{matrix} \quad ,$$

we see from (5b) that F application is translated into premultiplication by matrix

$$F = \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix}$$

from which

(8) $$F^n = \begin{vmatrix} f.(n-1) & f.n \\ f.n & f.(n+1) \end{vmatrix}$$

follows (by mathematical induction).

<u>Remark</u> "Analytical extension" of (0) yields $f.(-1) = 1$. This value yields in (8) for F^0 the unit matrix, as it should. (End of Remark.)

With F denoting a matrix, the $\circ$ in (7) has to be interpreted as matrix multiplication; then, (7) and (8) yield — for the top-right element of F^{a+b} —

(9) $\quad f.(a+b) = f.(a-1) \cdot f.b + f.a \cdot f.(b+1)$,

which contains all the terms occurring in (4), which has to be proved. To do so, we observe

$\quad f.(a+b)$ <u>gcd</u> $f.b$
$=\quad \{(9)\}$
$\quad (f.(a-1) \cdot f.b + f.a \cdot f.(b+1))$ <u>gcd</u> $f.b$
$=\quad$ {property of <u>gcd</u>}
$\quad (f.a \cdot f.(b+1))$ <u>gcd</u> $f.b$
$=\quad$ {Lemma 0, below, with $n := b$; property <u>gcd</u>}
$\quad f.a$ <u>gcd</u> $f.b$.

<u>Lemma 0</u> $\qquad f.n$ <u>gcd</u> $f.(n+1) \;=\; 1$.

For $n = 0$, the lemma follows from the definition of $f.0$ and $f.1$. For larger values of n , it follows from Euclid's Algorithm with $X, Y := f.n, f.(n+1)$. On account of the last definition in (0)

$$\langle \underline{E} m : m > 0 : \{x, y\} = \{f.m, f.(m+1)\} \rangle$$

is then an invariant of the algorithm.

* * *

The transition from (0) to (5) could strike one as a rabbit, but it isn't for someone who has seen a little bit more. It underlies one of the oldest examples of program transformation —from the pen of R.M. Burstall— ; it is quite common wherever functional composition plays a significant rôle —functional programming, constructive type theory, or category theory, just to mention a few— . I am more surprised by the very different ways in which the gcd enters the picture.

Austin, 9 April 1990

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