Lambek and Moser revisited.

Let \( f \) be an ascending sequence of natural numbers, i.e.

\[
(\forall i,j: 0 \leq i < j: f(i) \leq f(j))
\]

that is unbounded, i.e.

\[
(\exists j \geq 0: (\exists i \geq 0: f(i) > j))
\]

The function \( \text{lambo} \) is defined as follows: \( \text{lambo}(\mathcal{P}) \) is a sequence of natural numbers and \( g = \text{lambo}(\mathcal{P}) \) means that for all \( y \geq 0 \), \( g(y) = x \), where \( x \) stands for the minimum value such that \( f(x) > y \), or, more formally, where \( x \) satisfies

\[
R: (\forall i: 0 \leq i < x: f(i) \leq y) \land f(x) > y
\]

In order to compute \( g \), we design a program with the invariant relation

\[
\mathcal{P}: (\forall i: 0 \leq i < x: f(i) \leq y) \land f(x) > y
\]

\[
x, y := 0, 0; \{\mathcal{P}\}
\]

\[
do\ f(x) = y \rightarrow x := x + 1 \{\mathcal{P}\}
\]

\[
\mid f(x) > y \rightarrow \{\mathcal{P}\} g(y) := x; y := y + 1 \{\mathcal{P}\}
\]

od
Note firstly, that the program
\[
\text{do } f(x) = y \rightarrow x := x + 1 \text{ od}
\] (2)
terminates because \( f \) is unbounded; note secondly, that the program
\[
\text{do } f(x) > y \rightarrow g(y) := x; y := y + 1 \text{ od}
\] (3)
terminates; note, finally, that on account of the last term of \( f \) program (1) fails to terminate.

Consider now program (4), in which we assume \( g \) to be initialized \( g = \lambda x.0(f(x)) \); the same \( f \) is again an invariant.

\[
x, y := 0, 0; \{ P \}
\]
\[
\text{do } f(x) = y \rightarrow x := x + 1 \{ P \}
\]
\[
\text{until } f(x) > y \rightarrow \{ \text{R, hence } g(y) = x \} y := y + 1 \{ P \}\]
\[
\text{od}
\]

Also program (4) fails to terminate; because we are entitled to assert \( g(y) = x \) in the second guarded command, the program still fails to terminate when we include that relation in the second guard

\[
x, y := 0, 0; \{ Q \}
\]
\[
\text{do } f(x) = y \rightarrow x := x + 1 \{ Q \}
\]
\[
\text{until } f(x) > y \land g(y) = x \rightarrow y := y + 1 \{ Q \}\]
\[
\text{od}
\]
From the fact that (5) fails to terminate we conclude a further invariant
\( \Theta : \quad g(y) \geq x \).

We conclude this by considering \( \neg \Theta : \quad g(y) < x \).
In that case (5) reduces to (2), of which \( \neg \Theta \) is obviously an invariant; because (2) terminates and (5) does not, \( \neg \Theta \) cannot occur. Having established the invariance of \( \Theta \), we conclude that the program still fails to terminate when we "strengthen" the first guard with \( \text{wp} ("x:=x+1", \Theta) \):

\[
\begin{align*}
\begin{array}{ll}
x, y := 0, 0; \\
do & \quad f(x) = y \land g(y) > x \rightarrow x := x + 1 \\
\quad \downarrow & \quad f(x) > y \land g(y) = x \rightarrow y := y + 1 \\
\od & \end{array}
\end{align*}
\]  
(6)

But program (6) is symmetric in the pairs \((x, f)\) and \((y, g)\); hence
\( (g = \text{lambo}(f)) = (f = \text{lambo}(g)) \),
in other words: the function \( \text{lambo} \) is its own inverse.

Finally, let us consider the program

\[
\begin{align*}
\begin{array}{ll}
x, y, n := 0, 0, 0; \\
do & \quad f(x) = y \land g(y) > x \rightarrow \{ x + f(x) = n \} \quad x, n := x + 1, n + 1 \\
\quad \downarrow & \quad f(x) > y \land g(y) = x \rightarrow \{ y + g(y) = n \} \quad y, n := y + 1, n + 1 \\
\od & \end{array}
\end{align*}
\]  
(7)
Program (7) has the obvious invariant \( x+y=n \), which justifies the two assertions. They, however, are the respective weakest preconditions for the invariance of

\[ Q_1: \text{the sets } \{i+f(i) \mid 0 \leq i < x\} \text{ and } \{j+g(j) \mid 0 \leq j < y\} \text{ form a partitioning of the first } n \text{ natural numbers } 0 \text{ through } n-1. \]

From the fact that \( x \) and \( y \) are unbounded and from the invariance of \( Q_1 \), which is true at initialization, we conclude the second result of Lambeck and Moser, viz. that \( \{i+f(i) \mid 0 \leq i \leq x\} \text{ and } \{j+g(j) \mid 0 \leq j \leq y\} \text{ form a partitioning of the natural numbers.} \)

\[ \text{Note. By introducing } n=x+y \text{ in program (7), we could have derived the second result of Lambeck and Moser immediately; it, in turn, implies that the function lambo is its own inverse. But I thought the independent derivation of (6) more fun. (End of Note.)} \]

* * *

I am not quite clear about the moral of the above. We have proved theorems about the function lambo by first deriving a program for it and then massaging the program. For me this is a novel
application of semantics preserving program transformations, and this novelty—as all such novelties—causes some mild excitement. On the other hand we know that a chain of program transformations is so close to a mechanically verifiable proof that it seems vain to hope to prove any "deep" theorems this way. (Here I should add that I get less and less certain about the significance of the supposed difference between "deep" and "shallow" theorems.) It is possibly no more than an occasionally neat way of formulating an otherwise not unusual mathematical argument.

Plataanstraat 5 8 February 1981
5671 AL NUENEN prof. dr. Edsger W. Dijkstra
The Netherlands Burroughs Research Fellow.