# Lambek and Moser revisited.

Let $f$ be an ascending sequence of natural numbers, i.e.

$$(\underline{A}\, i,j : 0 \leq i < j : f(i) \leq f(j))$$

that is unbounded, i.e.

$$(\underline{A}\, j : j \geq 0 : (\underline{E}\, i : i \geq 0 : f(i) > j)) \quad .$$

The function lambo is defined as follows: lambo($f$) is a sequence of natural numbers and $g$ = lambo($f$) 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

$$R: \quad (\underline{A}\, i : 0 \leq i < x : f(i) \leq y) \wedge f(x) > y \quad .$$

In order to compute $g$, we design a program with the invariant relation

$$P: \quad (\underline{A}\, i : 0 \leq i < x : f(i) \leq y) \wedge f(x) \geq y$$

```
x, y := 0, 0 ; {P}                                          (1)
do  f(x) = y  →  x := x+1  {P}
 ‖  f(x) > y  →  {R} g(y) := x ; y := y+1  {P}
od
```

Note firstly, that the program

$$\underline{\text{do}}\ f(x)=y \rightarrow x := x+1\ \underline{\text{od}} \qquad (2)$$

terminates because $f$ is unbounded; note secondly, that the program

$$\underline{\text{do}}\ f(x)>y \rightarrow g(y) := x\ ;\ y := y+1\ \underline{\text{od}} \qquad (3)$$

terminates; note, finally, that on account of the last term of $P$ program (1) fails to terminate.

Consider now program (4), in which we assume $g$ to be initialized $g = \text{lambo}(f)$; the same $P$ is again an invariant.

$$\begin{array}{l} x, y := 0, 0\ ;\ \{P\} \qquad (4) \\ \underline{\text{do}}\ \ f(x)=y \rightarrow x := x+1\ \{P\} \\ \ \ [\!]\ \ f(x)>y \rightarrow \{R, \text{hence } g(y)=x\}\ y := y+1\ \{P\} \\ \underline{\text{od}} \end{array}$$

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

$$\begin{array}{l} x, y := 0, 0\ ;\ \{Q\} \qquad (5) \\ \underline{\text{do}}\ \ f(x)=y \rightarrow x := x+1\ \{Q\} \\ \ \ [\!]\ \ f(x)>y \wedge g(y)=x \rightarrow y := y+1\ \{Q\} \\ \underline{\text{od}} \end{array}$$

From the fact that (5) fails to terminate we conclude a further invariant

$Q$: $\quad g(y) \geq x$ .

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

```
x, y := 0, 0 ;
do f(x) = y ∧ g(y) > x → x := x+1                    (6)
 [] f(x) > y ∧ g(y) = x → y := y+1
od
```

But program (6) is symmetric in the pairs $(x, f)$ and $(y, g)$; hence

$$(g = \text{lambo}(f)) = (f = \text{lambo}(g)) \quad ,$$

in other words: the function lambo is its own inverse.

Finally, let us consider the program

```
x, y, n := 0, 0, 0 ;                                 (7)
do f(x) = y ∧ g(y) > x → { x + f(x) = n } x, n := x+1, n+1
 [] f(x) > y ∧ g(y) = x → { y + g(y) = n } y, n := y+1, n+1
od
```

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

Q1: the sets $\{i+f(i) \mid 0 \le i < x\}$ and $\{j+g(j) \mid 0 \le j < y\}$ form a partioning of the first n natural numbers 0 through n-1 .

From the fact that x and y are unbounded and from the invariance of Q1, which is true at initialization, we conclude the second result of Lambek and Moser, viz. that $\{i+f(i) \mid 0 \le i\}$ and $\{j+g(j) \mid 0 \le j\}$ form a partioning of the natural numbers.

Note. By introducing $n = x+y$ in program (1) we could have derived the second result of Lambek 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
5671 AL NUENEN
The Netherlands

8 February 1981

prof. dr. Edsger W. Dijkstra
Burroughs Research Fellow.