The list unfold

The list unfold, $unfoldl$, is specified by

$$\begin{array}{lcl} unfoldl & : & (a \rightsquigarrow \mathsf{Bool}) \rightsquigarrow (a \rightsquigarrow b) \rightsquigarrow (a \rightsquigarrow a) \rightsquigarrow a \rightsquigarrow [b] \\ unfoldl.p.f.g.x & = & \textbf{if } p.x \textbf{ then } [\,] \textbf{ else } f.x : unfoldl.p.f.g.(g.x) \end{array}$$

In this note we derive a procedure to compute $unfoldl$.

* *
*

As $unfoldl$ is a recurrence, we will compute it with a loop. For $ys : [b]$ and $X : a$ the loop's postcondition is

$$ys = unfoldl.p.f.g.X \quad .$$

We choose as invariant

$$unfoldl.p.f.g.X \;=\; ys \;{+\!\!+}\; unfoldl.p.f.g.x$$

which is established with $ys, x \;:=\; [\,], X$.
With regards to maintenance of the invariant we observe

$$\begin{array}{ll} & |[\ \text{Context:} \quad p.x \quad \Rightarrow \quad unfoldn.p.f.g.x \;=\; [\,] \\ & \quad ys \;{+\!\!+}\; unfoldn.p.f.x \\ = & \quad \{\ p.x\ \} \\ & \quad ys \\ &]| \end{array}$$

and

$$\begin{array}{ll} & |[\ \text{Context:} \quad \neg p.x \quad \Rightarrow \quad unfoldn.p.f.g.x \;=\; f.x : unfoldl.p.f.g.(g.x) \\ & \quad ys \;{+\!\!+}\; unfoldn.p.f.x \\ = & \quad \{\ \neg p.x\ \} \\ & \quad ys \;{+\!\!+}\; f.x : unfoldl.p.f.g.(g.x) \\ = & \quad \{\ \text{property of } :\ \} \\ & \quad ys \;{+\!\!+}\; [f.x] \;{+\!\!+}\; unfoldl.p.f.g.(g.x) \\ = & \quad \bigtriangledown\ ys := ys \;{+\!\!+}\; [f.x] \quad ; \quad x := g.x\ \bigtriangledown \\ & \quad ys \;{+\!\!+}\; unfoldn.p.f.x \end{array}$$

]|

$$\begin{array}{ccc} & * & * \\ & * & \end{array}$$

We have derived the procedure

$$\begin{array}{lll} unfoldl.p.f.g.X & = & |[\textbf{var } x : a;\ ys : [b]; \\ & & \quad x := X;\ ys := [\,]; \\ & & \quad \textbf{do } \neg p.n\ \rightarrow \\ & & \qquad ys := ys\ +\!\!+\ [f.x] \\ & & \quad\ ; x := g.x \\ & & \quad \textbf{od}; \\ & & \quad ys \\ & &]| \quad . \end{array}$$

So concludes our derivation.

2008.03.25

E. Emmanuel Macaulay
eric@mathmeth.com