The natural unfold

(*filed under "general programming techniques"*)

A natural unfold, h , is a recursive function of the shape

$$h = unfold.p.f \quad \equiv \quad h.n \;=\; \textbf{if } p.n \textbf{ then } 0 \textbf{ else } 1 + h.(f.n)$$

In this note we will derive a procedure, $unfoldn$, which computes a natural unfold.

* *
*

Specification

Let $p : a \rightsquigarrow \mathsf{Bool}$, $f : a \rightsquigarrow a$ and $n : a$. We specify $unfoldn$ by

- $unfoldn.p.f.n \;=\; \textbf{if } p.n \textbf{ then } 0 \textbf{ else } 1 + unfoldn.p.f.(f.n)$

- $[\, wp.(unfoldn.p.f.n).R \;\equiv\; R \,]$ for all R .

* *
*

Implementation

As $unfoldn$ is recursive, we will use a loop to compute it. For $r : \mathsf{Nat}$, $N : a$ the postcondition of our loop is

$$r = unfoldn.b.f.N \quad .$$

We choose as loop invariant

$$unfoldn.b.f.N \;=\; r + unfoldn.b.f.n$$

which we establish with $r := 0 \;;\; n := N$.

With regards to maintenance of the invariant we observe

$$
\begin{array}{ll}
|[\text{ Context:} & b.n \quad \Rightarrow \quad unfoldn.b.f.n \;=\; 0 \\
& r + unfoldn.b.f.n \\
= & \quad \{\; b.n \;\} \\
& r \\
]|
\end{array}
$$

and

$$
\begin{array}{ll}
|[\text{ Context:} & \neg b.n \quad \Rightarrow \quad unfoldn.b.f.n \;=\; 1 + unfoldn.b.f.(f.n) \\
& r + unfoldn.b.f.n \\
= & \quad \{\; \neg b.n \;\} \\
& r + 1 + unfoldn.b.f.(f.n) \\
= & \quad \bigtriangledown\; r := r + 1 \;;\; n := f.n \;\bigtriangledown \\
& r + unfoldn.b.f.n \\
]| & \quad .
\end{array}
$$

We have derived the following procedure

$$
\begin{array}{lll}
unfoldn.p.f.N & = & |[\mathbf{var}\; n : a;\; r : \mathsf{Nat}; \\
& & \quad r := 0;\; n := N; \\
& & \quad \mathbf{do}\; \neg b.n \;\rightarrow \\
& & \qquad r := r + 1 \\
& & \quad ; n := f.n \\
& & \quad \mathbf{od}; \\
& & \quad r \\
& &]| \quad .
\end{array}
$$

So concludes our derivation.

* *
*

Example: The Linear Search

For $b : \mathsf{Nat} \rightsquigarrow \mathsf{Bool}$, the Linear Search is a standard algorithm for finding the least x such that $b.x$, given that such an x exists. For $i : \mathsf{Nat}$ a program for the Linear Search is

$$
\begin{array}{l}
|[\mathbf{con}\ b : \mathsf{Nat} \rightsquigarrow \mathsf{Bool};\ \mathbf{var}\ x : \mathsf{Nat};\\
\quad \{\langle \exists\, i\ ::\ b.i \rangle\}\\
\quad x := unfoldn.b.(1+).0\\
\quad \{b.x \wedge \langle \forall\, i\ : i < x :\ \neg b.i \rangle\}\\
]|\quad .
\end{array}
$$

2008.03.17

E. Emmanuel Macaulay
eric@mathmeth.com