Triggered by JAW99

Eric Macaulay

0

This program computes a representation of natural n in base b, $2 \leqslant b$, and was written because I was not satisfied with the derivation of the algorithm presented in JAW99.

1

A representation of a number is a list of digits. We define a function *repr* which constructs a list with the invariant property that each element is less than b.

```
type Base = Int
type List a = [a]
type Digit = Int
repr :: Base → Int → List Digit
"Compute representation"
```

2

We identify two cases for n: n is less than b and b is at most n.

```
%
"Compute representation" ⊇
repr b n
    | n < b = "Representation if n is less than the base"
    | b ⩽ n = "Representation if n exceeds the base"
```

Note that here function application is denoted by juxtaposition.

3

The first case is straightforward:

$$\begin{array}{l} \% \\ \text{“Representation if } \ n \ \text{ is less than base”} \supseteq \\ \quad [n] \end{array}$$

4

If $b \leqslant n$ then we divide n by b and append the remainder to the representation for the quotient:

$$\begin{array}{l} \% \\ \text{“Representation if } \ n \ \text{ exceeds the base”} \supseteq \\ \quad repr\ b\ (n\ \mathbf{div}\ b) +\!\!+ [n\ \mathbf{mod}\ b] \end{array}$$

5

A hylo-program [1] to compute $repr$ is

$$\begin{array}{l} repr\ b = unfold\ pfg \\ \quad \mathbf{where}\ pfg\ n \\ \qquad \mid n < b = Left\ n \\ \qquad \mid b \leqslant n = Right\ (n\ \mathbf{div}\ b, n\ \mathbf{mod}\ b) \end{array}$$

Acknowledgement

I thank Jeremy Gibbons for the hint which enabled me to derive the unfold for $repr$.

[1] A hylo-program is a program of the form $fold\ f \cdot unfold\ pfg$ where $unfold$ is defined by

$$\begin{array}{l} unfold\ pfg\ x = \mathbf{case}\ pfg\ x\ \mathbf{of} \\ \quad Left\ y \quad \rightarrow [y] \\ \quad Right\ (x', y) \rightarrow unfold\ pfg\ x' +\!\!+ [y] \end{array}$$