

Project Euler, Problem 1

Find the sum of all natural multiples of 3 or 5 below 1000.

* *
*

Let the raised infix dot, \cdot , denote function composition. We are heading for a program of the form

$\text{foldr.f.e} \cdot \text{unfoldr.step}$.

The function `foldr` is defined by

$\text{foldr} \quad : \quad (a \rightarrow b \rightarrow b) \rightarrow b \rightarrow \text{List.a} \rightarrow b$
 $\text{foldr.f.e.[]} \quad = \quad e$
 $\text{foldr.f.e.}(x : xs) \quad = \quad f.x.(\text{foldr.f.e.}xs)$

Regarding `unfoldr`, recall the standard type `Maybe`:

datatype `Maybe(a : Set)` ::=
 `Nothing` : `Maybe a`
 `Just` : `a` \rightarrow `Maybe a`

The definition of `unfoldr` is

$\text{unfoldr} \quad : \quad (b \rightarrow \text{Maybe}(a \times b)) \rightarrow b \rightarrow \text{List.a}$
 $\text{unfoldr.step.x} \quad = \quad \text{case step.x of}$
 $\text{Just}(y, x') \rightarrow y : \text{unfoldr.step.x}'$
 $\text{Nothing} \rightarrow []$

* *
*

Our algorithm will proceed in four phases:

0. Generate the list $[i : 0 \leq i < 1000]$.
1. Filter out the non-multiples of 3.
2. Filter out the non-multiples of 5.
3. Compute the sum of the elements of the list.

* *
*

Generating the list

We generate our list with

`unfoldr.step.0` where `step = $\langle \lambda x \rightarrow \text{if } x = 1000 \text{ then Nothing else Just}(x, x + 1) \rangle$`

* *
*

Filtering multiples of 3 and 5

To remove non-multiples of 3 and 5 we use the standard function `filter` defined by

$$\begin{aligned} \text{filter} & : (a \rightarrow \text{Bool}) \rightarrow \text{List}.a \rightarrow \text{List}.a \\ \text{filter}.p & = \text{foldr}.g.[] \quad \text{where } g.x = \text{if } p.x \text{ then}(x :) \text{ else id} \end{aligned}$$

We define

$$\begin{aligned} \text{multiple3or5} & : \text{Int} \rightarrow \text{Bool} \\ \text{multiple3or5 } n & = n \bmod 3 = 0 \vee n \bmod 5 = 0 \end{aligned}$$

giving us as our filter

$$\text{filter}.multiple3or5$$

* *
*

Summing the list

To sum the list we define

$$\text{sum} = \text{foldr}.(+).0$$

* *
*

Our program is thus

$$\text{sum} \cdot \text{filter}.multiple3or5 \cdot \text{unfoldr}.step.0$$

* *
*

If we postulate that our program notation has non-strict semantics then the list need never exist as a whole. It remains advantageous, however, to fuse the phases into one in order to reduce the amount of heap space turnover.

We will first combine `sum` and `filter.multiple3or5`. To this end we recall the fusion theorem:

[[Context: f is strict $\wedge f.a = b \wedge f.(g.x.y) = h.x.(f.y)$

$$f \cdot \text{foldr}.g.a = \text{foldr}.h.b$$

]]

Let us solve the equation

$$h, b \quad :: \quad \text{sum} \cdot \text{filter}.p = \text{foldr}.h.b$$

Looking at the conditions of the fusion theorem, the first is satisfied by having `sum` be strict, and the second by taking `0 = b`. As for the third, we observe

$$\begin{aligned} & \text{sum}.(g.x.xs) = h.x.(\text{sum}.xs) \\ = & \quad \{ g \} \\ & \begin{cases} \text{sum}.(x : xs) = h.x.(\text{sum}.xs) & \text{if } \text{multiple3or5}.x \\ \text{sum}.xs = h.x.(\text{sum}.xs) & \text{if } \neg \text{multiple3or5}.x \end{cases} \\ \Leftarrow & \quad \{ \text{property of sum; Leibniz} \} \\ & \begin{cases} h.x = (x+) & \text{if } \text{multiple3or5}.x \\ h.x = \text{id} & \text{if } \neg \text{multiple3or5}.x \end{cases} \\ & \quad \quad \quad * \quad * \\ & \quad \quad \quad * \end{aligned}$$

It remains to fuse `foldr.h.0` and `unfoldr.step` by solving the equation

$$f \quad :: \quad f = \text{foldr}.h.0 \cdot \text{unfoldr}.step$$

We observe

$$\begin{aligned} & f.x \\ = & \quad \{ f \} \\ & \text{foldr}.h.0.(\text{unfoldr}.step.x) \\ = & \quad \{ \text{step}.x = \text{Nothing} \} \\ & \text{foldr}.h.0.[] \\ = & \quad \{ \text{foldr} \} \\ & 0 \end{aligned}$$

and

$$\begin{aligned} & f.x \\ = & \quad \{ f \} \\ & \text{foldr}.h.0.(\text{unfoldr}.step.x) \\ = & \quad \{ \text{step}.x = \text{Just}.(x, x + 1) \} \\ & \text{foldr}.h.0.(x : \text{unfoldr}.step.(x + 1)) \\ = & \quad \{ \text{foldr}; h \} \\ & \begin{cases} x + \text{foldr}.h.0.(\text{unfoldr}.step.(x + 1)) & \text{if } \text{multiple3or5}.x \\ \text{foldr}.h.0.(\text{unfoldr}.step.(x + 1)) & \text{if } \neg \text{multiple3or5}.x \end{cases} \\ = & \quad \{ f \} \\ & \begin{cases} x + f.(x + 1) & \text{if } \text{multiple3or5}.x \\ f.(x + 1) & \text{if } \neg \text{multiple3or5}.x \end{cases} \end{aligned}$$

* *
*

Our solution is thus f.0 or 233168.

Eric Macaulay
2008.05.19