2008 Formosan Summer School on Logic, Language, and Computation

Program Construction and Reasoning
Exercises for Day 2

Shin-Cheng Mu

July 7th, 2008

1 In-Class Exercises

1.1 Folds and Fold-Fusion

1. Given functions $f :: \alpha \rightarrow \beta$ and $g :: \alpha \rightarrow \gamma$, $split\, f\, g :: \alpha \rightarrow (\beta, \gamma)$ is a function defined by:

$$split\, f\, g\, a = (f\ a, g\ a).$$

Recall the definition of *steep* and *sum*. The definition of *steepsum* can be re-written as:

$$steepsum \quad = \quad split\ steep\ sum.$$

Also recall that the identity function *id* on lists is a fold: $id = foldr\,(:)\,[\,]$. Use the fold-fusion theorem to fuse $steepsum \cdot id$ into one fold.

2. Recall the definition of *scanr* from the lecture:

$$scanr\, f\ e \quad = \quad map\,(foldr\, f\ e) \cdot tails$$

and its implementation as a fold:

$$\begin{array}{l} scanr\, f\ e = foldr\,(sc\, f)\,[e] \\ \qquad\quad \textbf{where}\ sc\, f\ x\,(y{:}ys) = f\ x\ y : y : ys \end{array}$$

(a) Expand $scanr\,(+)\,0\,[1,2,3]$ step by step:

$$\begin{array}{ll} & scanr\,(+)\,0\,[1,2,3] \\ = & foldr\,(sc\,(+))\,[0]\,[1,2,3] \\ = & \ldots \end{array}$$

(b) Derive the implementation of $scanr\, f\ e$ by fusing $map\,(foldr\, f\ e) \cdot tails$ into one fold.

2 Take-Home Exercise (Due Date: July 10th)

You do not have to do the exercises below if you have completed any of the exercises from Day 1. Exercise 1 is worth 40 points while exercise 2 is worth 50 points.

1. The function $\mathit{filter}\ p$ selects from a list all elements satisfying a predicate p. For example, $\mathit{filter}\ \mathit{even}\ [1,2,3,4] = [2,4]$.

   (a) Give a recursive definition of filter:

   $$\begin{array}{lcl} \mathit{filter}\ p\ [\,] & = & \ldots \\ \mathit{filter}\ p\ (x{:}xs) & = & \ldots \end{array}$$

   (b) Define $\mathit{filter}\ p$ in terms of foldr.

   (c) Prove, by fold-fusion, that

   $$\mathit{filter}\ p \cdot \mathit{map}\ f \quad = \quad \mathit{map}\ f \cdot \mathit{filter}\ (p \cdot f).$$

   Hint: apply fold-fusion on both sides, and show that they are equal to the same fold.

2. Given two functions h_1 and h_2, the function $\mathit{split}\ h_1\ h_2$ computes the pair of their results:

   $$\mathit{split}\ h_1\ h_2\ xs \quad = \quad (h_1\ xs, h_2\ xs).$$

   In the special case when both h_1 and h_2 are defined by foldr:

   $$\begin{array}{lcl} h_1 & = & \mathit{foldr}\ f_1\ e_1, \\ h_2 & = & \mathit{foldr}\ f_2\ e_2, \end{array}$$

   the following "banana-split" rule allows us to express $\mathit{split}\ h_1\ h_2$ using one single foldr:

   $$\begin{array}{l} \mathit{split}\ h_1\ h_2 = \mathit{foldr}\ g\ (e_1, e_2), \\ \qquad\qquad \textbf{where}\ g\ x\ (y,z) = (f_1\ x\ y, f_2\ x\ z). \end{array}$$

   It optimises two traversal through the list to only one traversal. It is called "banana-split" because folds used to be written using a notation called "banana brackets".

   (a) The function $\mathit{split}\ \mathit{sum}\ \mathit{length}$ return the pair of sum and length of the input list. Use the banana-split rule to express $\mathit{split}\ \mathit{sum}\ \mathit{length}$ by a fold.

   (b) Prove the banana-split rule by fold fusion. Hint: recall that $\mathit{split}\ h_1\ h_2 = \mathit{split}\ h_1\ h_2 \cdot \mathit{id}$, and id is a fold.