Functional Programming
Exercise 3: Inductively Defined Functions

Shin-Cheng Mu

2012 Formosan Summer School on Logic, Language, and Computation
Aug 27 – Sep 7, 2012

1. Knowing that how addition on natural numbers can be defined, how does one define multiplication? Define a function $mul : int \to int \to int$ that performes multiplication, assuming both arguments are natural numbers. You may reuse $(+)$.

Solution:

$$
\begin{array}{l}
\textbf{let } mul\ m\ n \ = \ \textbf{match } m \textbf{ with} \\
\quad \mid 0 \to 0 \\
\quad \mid \mathbf{1}{+}\ m \to n + mul\ m\ n
\end{array}
$$

2. Define your version of the function $length : {}'a\ list$ that returns the length of a list (note that $[\,]$ has length 0).

Solution:

$$
\begin{array}{l}
\textbf{let rec } length \ = \ \textbf{function} \\
\qquad \mid\ [\,] \to 0 \\
\qquad \mid\ x :: xs \to \mathbf{1}{+}\ (length\ xs)
\end{array}
$$

3. Prove that $length$ distributes into $(@)$:

$$length\ (xs\ @\ ys) = length\ xs + length\ ys$$

Solution: Induction on xs. **Case** $xs := [\,]$:

$$
\begin{array}{ll}
 & length\ ([\,]\ @\ ys) \\
= & \{\ \text{definition of } (@)\ \} \\
 & length\ ys \\
= & \{\ \text{definition of } (+)\ \} \\
 & 0 + length\ ys \\
= & \{\ \text{definition of } length\ \} \\
 & length\ [\,] + length\ ys
\end{array}
$$

Case $xs := x :: xs$:

$$
\begin{aligned}
& \mathit{length}\ ((x :: xs) \mathbin{@} ys) \\
= \quad & \{\ \text{definition of } (@)\ \} \\
& \mathit{length}\ (x :: xs \mathbin{@} ys) \\
= \quad & \{\ \text{definition of } \mathit{length}\ \} \\
& \mathbf{1}{+}\ (\mathit{length}\ (xs \mathbin{@} ys)) \\
= \quad & \{\ \text{induction}\ \} \\
& \mathbf{1}{+}\ (\mathit{length}\ xs + \mathit{length}\ ys) \\
= \quad & \{\ \text{definition of } (+)\ \} \\
& (\mathbf{1}{+}\ \mathit{length}\ xs) + \mathit{length}\ ys \\
= \quad & \{\ \text{definition of } \mathit{length}\ \} \\
& \mathit{length}\ (x :: xs) + \mathit{length}\ ys
\end{aligned}
$$

4. Prove: $\mathit{sum} \ll \mathit{concat} = \mathit{sum} \ll \mathit{map}\ \mathit{sum}$.

Solution: This is equivalent to prove that for all xss, $\mathit{sum}\ (\mathit{concat}\ xss) = \mathit{sum}(\mathit{map}\ \mathit{sum}\ xss)$. We perform induction on xss.

Case $xss := [\,]$:

$$
\begin{aligned}
& \mathit{sum}\ (\mathit{concat}\ [\,]) \\
= \quad & \{\ \text{definition of } \mathit{concat}\ \} \\
& \mathit{sum}\ [\,] \\
= \quad & \{\ \text{definition of } \mathit{map}\ \} \\
& \mathit{sum}\ (\mathit{map}\ \mathit{sum}\ [\,])
\end{aligned}
$$

Case $xss := xs :: xss$:

$$
\begin{aligned}
& \mathit{sum}\ (\mathit{concat}\ (xs :: xss)) \\
= \quad & \{\ \text{definition of } \mathit{concat}\ \} \\
& \mathit{sum}\ (xs \mathbin{@} \mathit{concat}\ xss) \\
= \quad & \{\ \text{since } \mathit{sum}\ (xs \mathbin{@} ys) = \mathit{sum}\ xs + \mathit{sum}\ ys\ \} \\
& \mathit{sum}\ xs + \mathit{sum}\ (\mathit{concat}\ xss) \\
= \quad & \{\ \text{induction}\ \} \\
& \mathit{sum}\ xs + \mathit{sum}\ (\mathit{map}\ \mathit{sum}\ xss) \\
= \quad & \{\ \text{definition of } \mathit{sum}\ \} \\
& \mathit{sum}\ (\mathit{sum}\ xs :: \mathit{map}\ \mathit{sum}\ xss) \\
= \quad & \{\ \text{definition of } \mathit{map}\ \} \\
& \mathit{sum}\ (\mathit{map}\ \mathit{sum}\ (xs :: xss))
\end{aligned}
$$

The lemma that $\mathit{sum}\ (xs \mathbin{@} ys) = \mathit{sum}\ xs + \mathit{sum}\ ys$ needs to be proved separately, by another induction on xs.

5. Prove: $take\ n\ xs \mathbin{@} drop\ n\ xs = xs$, for all n and xs.

Solution: Induction on n.

Case $n := 0$:

$$
\begin{aligned}
& take\ 0\ xs \mathbin{@} drop\ 0\ xs \\
= & \quad \{ \text{ definitions of } take \text{ and } drop \ \} \\
& [\,] \mathbin{@} xs \\
= & \quad \{ \text{ defintion of } (@) \ \} \\
& xs
\end{aligned}
$$

Case $n := \mathbf{1}{+}n$. We further distinguish the cases when $x := [\,]$:

$$
\begin{aligned}
& take\ (\mathbf{1}{+}n)\ [\,] \mathbin{@} drop\ (\mathbf{1}{+}n)\ [\,] \\
= & \quad \{ \text{ definitions of } take \text{ and } drop \ \} \\
& [\,] \mathbin{@} [\,] \\
= & \quad \{ \text{ defintion of } (@) \ \} \\
& [\,]
\end{aligned}
$$

and when $x := x :: xs$:

$$
\begin{aligned}
& take\ (\mathbf{1}{+}n)\ (x :: xs) \mathbin{@} drop\ (\mathbf{1}{+}n)\ (x :: xs) \\
= & \quad \{ \text{ definitions of } take \text{ and } drop \ \} \\
& x :: take\ n\ xs \mathbin{@} drop\ n\ xs \\
= & \quad \{ \text{ induction } \} \\
& x :: xs
\end{aligned}
$$

6. Define functions $inits$ and $tails$, both of type $'a\ list \to\ 'a\ list\ list$, such that the former returns all prefixes of a list, while the latter returns all suffixes of a list. E.g.

- $inits\ [1;2;3] = [[\,];[1];[1;2];[1;2;3]]$
- $tails\ [1;2;3] = [[1;2;3];[2;3];[3];[\,]]$

Hint: Notice that $[\,]$ is a prefix (suffix) of any list. Thus both $inits$ and $tails$ always return a list containing $[\,]$. In particular, $inits\ [\,] = tails\ [\,] = [[\,]]$.

Solution:

$$
\begin{aligned}
& \mathbf{let\ rec}\ inits\ = \ \mathbf{function} \\
& \quad \mid [\,] \to [[\,]] \\
& \quad \mid x :: xs \to [\,] :: map\ (x ::)\ (inits\ xs), \\
\\
& \mathbf{let\ rec}\ tails\ = \ \mathbf{function} \\
& \quad \mid [\,] \to [[\,]] \\
& \quad \mid x :: xs \to (x :: xs) :: tails\ xs,
\end{aligned}
$$

7. Define a function $\mathit{fan} :: {}'a \to {}'a\ \mathit{list} \to {}'a\ \mathit{list}\ \mathit{list}$ such that $\mathit{fan}\ x\ xs$ inserts x into the 0th, 1st... nth positions of xs, where n is the length of xs. For example:

$$\mathit{fan}\ 5\ [1;2;3;4] = [[5;1;2;3;4];[1;5;2;3;4];[1;2;5;3;4];[1;2;3;5;4];[1;2;3;4;5]]$$

Solution:

$$\begin{aligned}&\textbf{let}\ \mathit{fan}\ x\ =\ \textbf{function}\\ &\quad \mid [\,] \to [[x]]\\ &\quad \mid y :: ys \to (x :: y :: ys) :: \mathit{map}\ (\textbf{fun}\ zs \to y :: zs)\ (\mathit{fan}\ x\ ys)\end{aligned}$$

8. Define $\mathit{perms} :: {}'a\ \mathit{list} \to {}'a\ \mathit{list}\ \mathit{list}$ that returns all permutations of the input list. For example:

$$\mathit{perms}\ [1;2;3] = [[1;2;3];[2;1;3];[2;3;1];[1;3;2];[3;1;2];[3;2;1]]$$

You will need several auxiliary functions defined in the lectures and in the exercises.

Solution:

$$\begin{aligned}&\textbf{let}\ \mathit{perms}\ =\ \textbf{function}\\ &\quad \mid [\,] \to [[\,]]\\ &\quad \mid x :: xs \to \mathit{concat}\ (\mathit{map}\ (\mathit{fan}\ x)\ (\mathit{perms}\ xs))\end{aligned}$$