Programming Languages: Functional Programming
Midterm Examination

Oct. 26, 2023

1. (20 points) 假設 $anime :: \mathsf{List}\ (\mathsf{String}, (\mathsf{Int}, \mathsf{Int}))$ 是一些有名動畫與其上映起訖年份的列表，例如：

$$\begin{aligned} anime = [&(\texttt{" 彩虹小馬"}, \quad (2011, 2017)),\\ &(\texttt{" 小叮噹"}, \quad\ \ (1973, 2005)),\\ &(\texttt{" 科學小飛俠"}, (1972, 1980)),\\ &(\texttt{" 海綿寶寶"}, \quad (1999, 2020)),\\ &(\texttt{" 庫洛魔法使"}, (1998, 2000))] \ . \end{aligned}$$

又假設 $people :: \mathsf{List}\ (\mathsf{String}, \mathsf{Int})$ 是一些人名以及他們今年 (2023 年) 的年紀，例如 $[(\texttt{"Alice"}, 20), (\texttt{"Bob"}, 28), (\texttt{"Clare"}, 17), (\texttt{"Dan"}, 45)]$, 定義函數

$$\begin{aligned} allWatchedBy :: &\ \mathsf{List}\ (\mathsf{String}, (\mathsf{Int}, \mathsf{Int})) \to \mathsf{List}\ (\mathsf{String}, \mathsf{Int}) \to \\ &\mathsf{List}\ (\mathsf{String}, \mathsf{List}\ \mathsf{String}) \ , \end{aligned}$$

使得 $allWatchedBy\ anime\ people$ 列出每部動畫看過的人。例如，在上述例子中，$allWatchedBy\ anime\ people$ 的結果是

$$\begin{aligned} [&(\texttt{" 彩虹小馬"}, \quad [\texttt{"Alice"}, \texttt{"Bob"}, \texttt{"Clare"}]),\\ &(\texttt{" 小叮噹"}, \quad\ \ [\texttt{"Bob"}, \texttt{"Dan"}]),\\ &(\texttt{" 科學小飛俠"}, [\,]),\\ &(\texttt{" 海綿寶寶"}, \quad [\texttt{"Alice"}, \texttt{"Bob"}, \texttt{"Clare"}]),\\ &(\texttt{" 庫洛魔法使"}, [\texttt{"Bob"}])] \ . \end{aligned}$$

提示：你可能用得上函數 $(\downarrow) :: \mathsf{Int} \to \mathsf{Int} \to \mathsf{Int}$, 傳回兩個參數之中的較小者，或 $(\uparrow) :: \mathsf{Int} \to \mathsf{Int} \to \mathsf{Int}$, 傳回兩個參數之中的較大者。

Solution: 以下是一種可能寫法。

$$\begin{aligned} &allWatchedBy :: \mathsf{List}\ (\mathsf{String}, (\mathsf{Int}, \mathsf{Int})) \to \mathsf{List}\ (\mathsf{String}, \mathsf{Int}) \to \\ &\qquad\qquad \mathsf{List}\ (\mathsf{String}, \mathsf{List}\ \mathsf{String})\\ &allWatchedBy\ anime\ people = \\ &\quad map\ (\lambda(prog, duration) \to (prog, watchedBy\ duration\ people))\ anime \ , \\ &watchedBy :: (\mathsf{Int}, \mathsf{Int}) \to \mathsf{List}\ (\mathsf{String}, \mathsf{Int}) \to \mathsf{List}\ \mathsf{String}\\ &watchedBy\ duration = map\ fst \cdot filter\ (overlap\ duration \cdot snd) \end{aligned}$$

$$\textbf{where } overlap\ (x,y)\ age = y \downarrow w > x \uparrow z$$
$$\textbf{where } birth = 2023 - age$$
$$(z,w) = (birth + 3, birth + 18)\ .$$

2. (20 points) 證明 *map*-fusion 定理：對所有 f 與 g, $map\ f \cdot map\ g = map\ (f \cdot g)$.

Solution: That is equivalent to proving that for all xs, $map\ f\ (map\ g\ xs) = map\ (f \cdot g)\ xs$. The proof is an induction on xs.

Case $xs := [\,]$:

$$\begin{aligned}
& map\ f\ (map\ g\ [\,]) \\
= \quad & \{ \text{ definition of } map \} \\
& [\,] \\
= \quad & \{ \text{ definition of } map \} \\
& map\ (f \cdot g)\ [\,]\ .
\end{aligned}$$

Case $xs := x : xs$:

$$\begin{aligned}
& map\ f\ (map\ g\ (x : xs)) \\
= \quad & \{ \text{ definition of } map \} \\
& map\ f\ (g\ x : map\ g\ xs) \\
= \quad & \{ \text{ definition of } map \} \\
& f\ (g\ x) : map\ f\ (map\ g\ xs) \\
= \quad & \{ \text{ induction} \} \\
& f\ (g\ x) : map\ (f \cdot g)\ xs \\
= \quad & \{ \text{ definition of } (\cdot) \} \\
& (f \cdot g)\ x : map\ (f \cdot g)\ xs \\
= \quad & \{ \text{ definition of } map \} \\
& map\ (f \cdot g)\ (x : xs)\ .
\end{aligned}$$

3. (15 points) Haskell 標準函式庫中有個 *zipWith* 函數，型別為 $zipWith :: (a \to b \to c) \to \text{List}\ a \to \text{List}\ b \to \text{List}\ c$. 給定 f, $zipWith\ f\ xs\ ys$ 將 xs 與 ys 中位置相對應的元素丟給 f。例如 $zipWith\ (+)\ [1,2,3,4]\ [5,6] = [6,8]$; $zipWith\ (:)\ [1,2]\ [[4,5],[],[7,8]] = [[1,4,5],[2]]$. 由以上兩例可發現，當兩個串列長度不同時，$zipWith\ f$ 把多出的元素捨棄掉。

請定義一個函數 *lzipWith*，行為類似 *zipWith*, 但會把多出的元素留下來。例如

$$lzipWith\ (+)\ [1,2,3,4]\ [5,6] = [6,8,3,4]\ .$$

該定義需包括型別。

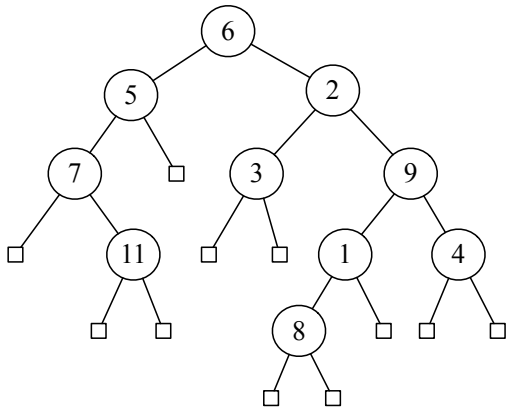

Figure 1: An internally labelled binary tree.

Solution:

$$
\begin{aligned}
&\mathit{lzipWith} :: (a \to a \to a) \to \mathsf{List}\ a \to \mathsf{List}\ a \to \mathsf{List}\ a \\
&\mathit{lzipWith}\ f\ [\,]\ \ \ \ \ \ ys\ \ \ \ \ \ = ys \\
&\mathit{lzipWith}\ f\ (x : xs)\ [\,]\ \ \ \ \ \ = x : xs \\
&\mathit{lzipWith}\ f\ (x : xs)\ (y : ys) = f\ x\ y : \mathit{lzipWith}\ f\ xs\ ys\ \ .
\end{aligned}
$$

4. (20 points) 下述資料結構表示標記在內部的二元樹：

$$\textbf{data}\ \mathsf{ITree}\ a = \mathsf{Null} \mid \mathsf{Node}\ a\ (\mathsf{ITree}\ a)\ (\mathsf{ITree}\ a)\ \ .$$

例如，t 可以畫成圖 1的二元樹。Node 為圖中圓形的節點，有標記；Null 是方形的，無標記。

```
t :: ITree Int
t = Node 6 (Node 5 (Node 7 Null
                           (Node 11 Null Null))
                   Null)
           (Node 2 (Node 3 Null Null)
                   (Node 9 (Node 1 (Node 8 Null Null) Null)
                           (Node 4 Null Null)))  .
```

請寫一個函數 $\mathit{levels} :: \mathsf{ITree}\ a \to \mathsf{List}\ (\mathsf{List}\ a)$, 傳回一顆樹每個水平層的標記。例如 $\mathit{levels}\ t = [[6],[5,2],[7,3,9],[11,1,4],[8]]$.

提示：可用課堂上提及過的，以及本考卷中定義過的函數。

Solution:

$$
\begin{aligned}
&\mathit{levels} :: \mathsf{ITree}\ a \to \mathsf{List}\ (\mathsf{List}\ a) \\
&\mathit{levels}\ \mathsf{Null} \qquad\quad = [\,] \\
&\mathit{levels}\ (\mathsf{Node}\ x\ t\ u) = [x] : \mathit{lzipWith}\ (+\!\!+)\ (\mathit{levels}\ t)\ (\mathit{levels}\ u)\ \ .
\end{aligned}
$$

5. 給定串列 xs, $\mathit{dels}\ xs$ 計算「從 xs 之中移除一個元素」的所有可能結果。例如 dels `"abcde"` $=$ [`"bcde"`, `"acde"`, `"abde"`, `"abce"`, `"abcd"`].

$$
\begin{aligned}
&\mathit{dels} :: \mathsf{List}\ a \to \mathsf{List}\ (\mathsf{List}\ a) \\
&\mathit{dels}\ [\,] \quad\ = ?? \\
&\mathit{dels}\ (x : xs) = xs : \mathit{map}\ (x:)\ (\mathit{dels}\ xs)\ \ .
\end{aligned}
$$

(a) (5 points) Base case 中 $\mathit{dels}\ [\,] = ??$ 的右手邊應該是什麼？

Solution: $\mathit{dels}\ [\,] = [\,]$.

(b) (20 points) 證明

$$\mathit{dels}\ (xs +\!\!+ ys) = \mathit{map}\ (+\!\!+ ys)\ (\mathit{dels}\ xs) +\!\!+ \mathit{map}\ (xs +\!\!+)\ (\mathit{dels}\ ys)\ \ .$$

提示：你可能用到幾個性質

- $([\,] +\!\!+) = \mathit{id}$, 其中 id 為 identity function, $\mathit{id}\ x = x$.
- $\mathit{map}\ \mathit{id} = \mathit{id}$.
- $\mathit{map}\ f\ (xs +\!\!+ ys) = ?? +\!\!+ ??$, 問號處自己完成。
- $(x:) \cdot (+\!\!+ [y]) = (+\!\!+ [y]) \cdot (x:)$.
- $((x : xs) +\!\!+) = ((x:)??)$, 問號處自己完成。
- 以及考卷中提及的性質。

這些性質可不用證明。

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

$$
\begin{aligned}
&\ \ \mathit{map}\ (+\!\!+ ys)\ (\mathit{dels}\ [\,]) +\!\!+ \mathit{map}\ ([\,] +\!\!+)\ (\mathit{dels}\ ys) \\
&= \quad \{\ \text{definition of } \mathit{map} \text{ and } \mathit{dels}\ \} \\
&\ \ [\,] +\!\!+ \mathit{map}\ ([\,] +\!\!+)\ (\mathit{dels}\ ys) \\
&= \quad \{\ ([\,] +\!\!+) = \mathit{id} \text{ and } \mathit{map}\ \mathit{id} = \mathit{id}\ \} \\
&\ \ \mathit{dels}\ ys \\
&= \quad \{\ \text{definition of } (+\!\!+)\ \} \\
&\ \ \mathit{dels}\ ([\,] +\!\!+ ys)\ \ .
\end{aligned}
$$

Case $xs := x : xs$:

$$
\begin{aligned}
& map\ (+\!\!+\ ys)\ (dels\ (x:xs)) +\!\!+ map\ ((x:xs) +\!\!+)\ (dels\ ys) \\
= \quad & \{ \text{ definition of } dels \} \\
& map\ (+\!\!+\ ys)\ (xs : map\ (x:)\ (dels\ xs)) +\!\!+ map\ ((x:xs) +\!\!+)\ (dels\ ys) \\
= \quad & \{ \text{ definition of } map \} \\
& (xs +\!\!+ ys) : map\ (+\!\!+\ ys)\ (map\ (x:)\ (dels\ xs)) +\!\!+ map\ ((x:xs) +\!\!+)\ (dels\ ys) \\
= \quad & \{ map\text{-fusion} \} \\
& (xs +\!\!+ ys) : map\ ((+\!\!+\ ys) \cdot (x:))\ (dels\ xs) +\!\!+ map\ ((x:xs) +\!\!+)\ (dels\ ys) \\
= \quad & \{ (+\!\!+\ ys) \cdot (x:) = (x:) \cdot (+\!\!+\ ys) \text{ and } ((x:xs) +\!\!+) = (x:) \cdot (xs +\!\!+) \} \\
& (xs +\!\!+ ys) : map\ ((x:) \cdot (+\!\!+\ ys))\ (dels\ xs) +\!\!+ map\ ((x:) \cdot (xs +\!\!+))\ (dels\ ys) \\
= \quad & \{ map\text{-fusion}, map\ f\ (xs +\!\!+ ys) = map\ f\ xs +\!\!+ map\ f\ ys \} \\
& (xs +\!\!+ ys) : map\ (x:)\ (map\ (+\!\!+\ ys)\ (dels\ xs) +\!\!+ map\ (xs +\!\!+)\ (dels\ ys)) \\
= \quad & \{ \text{ induction } \} \\
& (xs +\!\!+ ys) : map\ (x:)\ (dels\ (xs +\!\!+ ys)) \\
= \quad & \{ \text{ definition of } dels \} \\
& dels\ (x : (xs +\!\!+ ys)) \\
= \quad & \{ \text{ definition of } (+\!\!+) \} \\
& dels\ ((x:xs) +\!\!+ ys) \quad .
\end{aligned}
$$