Introduction to Functional Programming using Haskell

Errata

April 7, 1999

Chapter 1

page 3, line 1 Replace ? square 14198724 by ? square 14197824.

page 9, line 14 Replace “hit the the interrupt key” by “hit the interrupt key”.

page 22, line 8 Insufficient space between names in square square 3. [This unfortunates space compression occurs in various places throughout the text.]

page 23, line 14 “Look again the previous . . .” should read “Look again at the previous . . .”

page 25, line 22 Replace by “The link between the two is the requirement that the im-

plementation satisfies . . .”

Chapter 2

page 33, lines 1 - 8 The text is confused. There are two solutions:

Either replace it by “We can declare Bool to be an instance of Ord by writing

\[
\begin{align*}
\text{instance } \text{Ord } \text{Bool } \text{where} \\
\text{False } \leq \text{False } & = \text{ True} \\
\text{False } \leq \text{True } & = \text{ True} \\
\text{True } \leq \text{False } & = \text{ False} \\
\text{True } \leq \text{True } & = \text{ True} \\
\end{align*}
\]

The alternative definition, namely \( x \leq y = \text{not } x \lor y \), doesn’t quite work in the way expected (see Exercise 2.1.2). ”

Or replace lines 7 and 8 by “As an alternative definition we can write \( x < y = \text{not } x \land y \).

page 33, line 20 “Note that the two occurrences of . . .”

page 34 Delete Exercise 2.1.2.
In Exercise 2.1.9 replace the last sentence by “Show that these properties hold for the definition of (\equiv) on the well-defined values of Bool.

“a different entity from the decimal number 7;...”

“but does not depend ...”

Systematically interchange the names toEnum and fromEnum in the whole of Section 2.3.

Interchange toEnum and fromEnum.

The type of plus should be

$$plus : (\alpha \to \beta, \gamma \to \delta) \to Either \alpha \gamma \to Either \beta \delta$$

Note that Haskell uses the name either rather than case.

Chapter 3

Replace h (foldn h b (Succ n)) by h (foldn h b n).

Replace by

$$\text{Rat } x \text{ } y \equiv \text{Rat } u \text{ } v = (x \times v) \equiv (y \times u)$$

Replace sentence beginning “Among possible representations” with “Among possible representations we can choose one in which \(-5 \leq z < 5\) and abs y is as small as possible.

“since programs that avoid case analyses are clearer and simpler than those that do not, ...”

The definition of done should read

$$\text{done } (m, n) = (m + 1 \equiv n)$$

In the definition of y3 a division by 2 is missing:

$$y_3 = (1.4167 + 2/1.4167)/2 = 1.4142157$$
Chapter 4

page 103, line 14 First line in definition of \texttt{init} should read: \texttt{init} \([x] = []\).

page 112, line 10 “This equation is valid provided \(p\) and \(q\) are strict functions.”

page 114, line 4 Definition of \texttt{pyth} should read

\[ pyth (x, y, z) = (x \times x + y \times y \equiv z \times z) \]

page 116, line 14 Last line in definition of \texttt{zip} should read:

\[ \texttt{zip}(x:xs)(y:ys) = (x,y):\texttt{zip} xs ys \]

page 116, line 18 “the scalar product of two vectors \(x\) and \(y\) of size \(n\) is defined by . . .”

page 121, line 22 The function \texttt{zip} can be defined as an instance of \texttt{foldr}: we have \texttt{zip} = \texttt{foldr} \(f\) \(e\) where

\[
\begin{align*}
eys &= [] \\
f \times g [] &= [] \\
f \times g (y : ys) &= (x, y) : g ys
\end{align*}
\]

page 124, line 12 “the first is clearer, while the second is more efficient.”

page 124, line -6 The type of \texttt{scanl} should read:

\[ \texttt{scanl} :: (\beta \to \alpha \to \beta) \to \beta \to [\alpha] \to [\beta] \]

page 125, line -1 Replace \(a\) by \(e\) in equations involving \texttt{scanr}.

page 126, lines 9,10 Replace \(a\) by \(e\) in equations involving \texttt{scanr}.

page 127 Exercise 4.5.9 should read “What list does \texttt{scanl} (/) 1 [1..n] produce?”

page 125 In Exercise 4.5.11 the type of \texttt{convert} should be \texttt{Liste} \(\alpha \to [\alpha]\).

page 129, line -4 “Both sides simplify to \(x \oplus y\).”

page 129, line -1 Replace \(z;xs\) by \(z : xs\).

page 130, line -7 Missing period at end of paragraph.

page 137 In Exercise 4.6.10 the law should read

\[ \texttt{foldl1} (\oplus) \cdot \texttt{scanl} (\otimes) e = \texttt{fst} \cdot \texttt{foldl} (\otimes) (e, e) \]
Chapter 5

page 146, line 9 Replace assign xs by assign.

page 146, line 12 The definition of mktriple should read

\[
\text{mktriple} (xn, xm) xr = (xn, xm, xr)
\]

page 148, line 18 The last line of the definition of sortby should read:

\[
\text{sortby} f (x : y : xs) = \text{mergeby} f (\text{cross} (\text{sortby} f, \text{sortby} f) (\text{divide} (x : y : xs)))
\]

page 163, line 12 The three = signs on the right-hand side of the definition of leap should be == signs.

page 165, lines 12,15 The types of stackWith and spreadWith should be

\[
\begin{align*}
\text{stackWith} & :: \text{Height} \to [\text{Picture}] \to \text{Picture} \\
\text{spreadWith} & :: \text{Width} \to [\text{Picture}] \to \text{Picture}
\end{align*}
\]

page 167, line 16 Replace entries \((d, s)\) with just entries.

page 167, line 23 In the definition of dnames the conversion to type Picture is omitted, so prefix the right-hand side with row.

Chapter 6

page 185, lines -6 – -1 Replace definition of fork by

\[
\begin{align*}
\text{fork} & :: \text{Atree} \alpha \to \text{Atree} \alpha \to \text{Atree} \alpha \\
\text{fork} \ xt \ yt & = \text{Fork} (\text{lsize} \ xt) \ xt \ yt \\
\text{lsize} & :: \text{Atree} \alpha \to \text{Int} \\
\text{lsize} (\text{Leaf} \ x) & = 1 \\
\text{lsize} (\text{Fork} \ n \ xt \ yt) & = n + \text{lsize} \ yt
\end{align*}
\]

page 186, line 6 The two occurrences of mkBtree should be replaced by mkAtree.

page 187 In Exercise 6.1.3 the definition of subtrees should read

\[
\begin{align*}
\text{subtrees} & :: \text{Btree} \alpha \to [\text{Btree} \alpha] \\
\text{subtrees} (\text{Leaf} \ x) & = [\text{Leaf} \ x] \\
\text{subtrees} (\text{Fork} \ xt \ yt) & = [\text{Fork} \ xt \ yt] + \text{subtrees} \ xt + \text{subtrees} \ yt
\end{align*}
\]

page 188, line 9 The type of member should read

\[
\text{member} :: \text{Ord} \alpha \Rightarrow \alpha \to \text{Stree} \alpha \to \text{Bool}
\]

4
Space compressed in member \( x \) \( \times t \).

The type of \( \text{height} \) should read:

\[
\text{height} :: \text{Ord} \, \alpha \Rightarrow \text{Stree} \, \alpha \rightarrow \text{Int}
\]

The identity should read:

\[
x \ast s = x \ast [\text{head} \, s] \ast \text{tail} \, s
\]

Exercise 6.2.4 should read “Prove that \( \text{inordered} \, (\text{insert} \, x \, \times t) = \text{True} \) for all finite binary search trees \( x \times t \).

The type of \( \text{heapify} \) should read:

\[
\text{heapify} :: \text{Ord} \, \alpha \Rightarrow \text{Htree} \, \alpha \rightarrow \text{Htree} \, \alpha
\]

The type of \( \text{sift} \) should read:

\[
\text{sift} :: \text{Ord} \, \alpha \Rightarrow \alpha \rightarrow \text{Htree} \, \alpha \rightarrow \text{Htree} \, \alpha \rightarrow \text{Htree} \, \alpha
\]

Omit the local definition of \( \text{maxlist} \), and insert the following sentence in the text: “Recall that \( \text{maxlist} = \text{foldl1} \, (\text{max}) \).

The left-hand expression should read: \( f(g(x, y), z, h(t)) \).

The type of \( \text{combine} \) should read: \( \text{combine} :: [[\alpha]] \rightarrow [[\alpha]] \).

Same correction as above.

The type declaration of \( \text{CodeTable} \) should read:

\[
\text{type} \, \text{CodeTable} = [(\text{Char}, [\text{Bit}], \text{Int})]
\]

Replace local definition by \( \text{where} \, (y \times s, z \times s) = \text{span} \, (\equiv x) \, x \times s \).

Chapter 7

The type of \( \text{dfcat} \) should read:

\[
\text{dfcat} :: [\text{Rose} \, \alpha] \rightarrow [\alpha] \rightarrow [\alpha]
\]

The second line in the definition of \( \text{fills} \) should read:

\[
\text{fills} \, (w : ws) = [u : vss | (u, v) \leftarrow \text{splits} \, (w : ws); vss \leftarrow \text{fills} \, vss]
\]

Space compression in \( \text{fill} \, v \times s \).
Chapter 8

page 256, line 16 “The second implementation therefore has a different efficiency from the first,...”

page 256, line 21 The right-hand side of the second axiom for back should be
join x (back (join y xq))

page 272, lines 13,15 Remove closing parenthesis from right-hand expressions.

page 278, line 12 The type of fork should be
fork :: α → Htree α → Htree α

page 279, line 1 The type of delMin should be
delMin :: Ord α ⇒ Htree α → Htree α

page 279, line 3 The type of union should be
union :: Ord α ⇒ Htree α → Htree α → Htree α

page 280, line 6 The type of mkBag should be
mkBag :: Ord α ⇒ [α] → Htree α

page 280, line 8 The type of mkTwo should be
mkTwo :: Ord α ⇒ Int → [α] → (Htree α, [α])

page 284, line 9 Replace right-hand side of otherwise branch by
Fork n xt (update yt (k - m) x)

page 288, line 4 Replace nullys by null ys.

page 288, line 22 It should be pointed out that the definition of abstr is exactly the same as in the implementation of Section 8.1 since

ys ++ reverse xs = reverse (xs ++ reverse ys)

page 289, line 14 Replace reverseys by reverse ys.

page 290, line -1 Replace last line by
(3, 0, rot (rot [] [1]) [3, 2] [], [])

page 291, line 2 Replace by
(3, 3, rot (rot [] [1]) [3, 2] [], [6, 5, 4])

page 291, line 4 Replace by
(7, 0, rot (rot (rot [] [1]) [3, 2] []) [7, 6, 5, 4] [], [])

page 291, line 6 Replace by
(7, 7, rot (rot (rot [] [1]) [3, 2] []) [7, 6, 5, 4] [], [14, 13..8])
Chapter 9
page 296, line 17 “the computer determines the first four elements . . .”

Chapter 10
page 330, line 13 Should add “where C may be empty”.
page 342 In Exercise 10.2.2 the type definition should read

\[
\text{newtype } \text{Count } \alpha = \text{CNT } (\alpha, \text{Counter})
\]

Chapter 11
page 365, line -10 The definition should read

\[
p \mid \text{orelse } q = \text{MkP } f \\
\text{where } f s = \text{if null } ps \text{ then } \text{apply } q s \text{ else } ps \\
\text{where } ps = \text{apply } p s
\]

page 365, line -6 The operator \text{orelse} does not satisfy the distributive law of \text{plus}.

page 368, line -1 Replace \( \triangleright \) by \( \triangleright\triangleright \).

page 373 In Exercise 11.4.1 add “for deterministic parsers \( p \) and \( q \)”.

Chapter 12
page 384, line -11 Type of \text{notuple} should read

\[
\text{notuple} :: \text{Parser } [\text{Expr}]
\]

Appendix
page 411, line -1 Replace by \text{head} (\( x : xs \)) = \( x \).