Programming Languages: Functional Programming Practicals 8. Streams and Codata

Shin-Cheng Mu

Autumn, 2025

1. Try constructing the following streams. There might be more than one way to do so.
   (a) The stream of alternating 1 and -1's: $[1, -1, 1, -1\,..]$. Try generating it in a way different from that in the handouts.
   (b) The stream of all natural numbers ($[0, 1, 2\,..]$). Try generating it in a way different from that in the handouts.
   (c) All the square numbers: $[0, 1, 4, 9, 16, 25\,..]$.
   (d) The sequence of natural numbers divisible by 3.
   (e) The sequence of natural numbers *not* divisible by 3.
   (f) The sequence of all finite binary strings (having type Stream (List Int)):
   $$[[\,], [0], [1], [0,0], [1,0], [0,1], [1,1], [0,0,0], [1,0,0], [0,1,0]...] \ .$$
   (g) The bit-reversed representation of binary numbers (having type Stream (List Int)):
   $$[[\,], [1], [0,1], [1,1], [0,0,1], [1,0,1], [0,1,1], [1,1,1], [0,0,0,1]...] \ .$$
   (h) A sequence indexed by positive integers (starting from 1), such that position i is True if i is a power of two ($2, 4, 8, 16$.. etc), False otherwise.

Solution:

1. $1 \curlyvee (-1)$.
2. $\mathit{iterate}\ (1+)\ 0$.
3. $\mathit{nat} \times \mathit{nat}$, or $\mathit{map}\ (\lambda x \rightarrow x\,\hat{}\,2)\ \mathit{nat}$, etc.
4. $3 \times \mathit{nat}$.

5. $1 + 3 \times nat \curlyvee 2 + 3 \times nat$.

6. $allBits = [\,] : map\ (0{:})\ allBits \curlyvee map\ (1{:})\ allBits$.

7. $binReps = [\,] : (map\ (1{:})\ binReps) \curlyvee (map\ (0{:})\ (tail\ binReps))$. For an admissible definition, you might want

$$\begin{aligned} binReps\ &= [\,] : binReps' \\ binReps' &= [1] : map\ (0{:})\ binReps' \curlyvee map\ (1{:})\ binReps' \ . \end{aligned}$$

8. $pot =$ True : $pot \curlyvee repeat$ False.

2. Let n be an Int, prove that $n + xs = map\ (n+)\ xs$. **Note**: recall that, according to the definition in our handouts, in $n + xs$, n is actually $repeat\ n$ and $(+)$ is $zipWith\ (+)$.

Solution: Expanding $n + xs$:

$$\begin{aligned} & n + xs \\ = \quad & \{ \text{ definition of } repeat \} \\ & (n : repeat\ n) + xs \\ = \quad & \{ \text{ definition of } (+) \} \\ & (n + head\ xs) : (n + tail\ xs) \ , \end{aligned}$$

while $map\ (n+)\ xs = (n + head\ xs) : map\ (n+)\ (tail\ xs)$. Both have the form $f\ xs = (n + head\ xs) : f\ (tail\ xs)$, which is an admissible expression, and therefore $n + xs = map\ (n+)\ xs$.

3. To see the importance of admissibility, let us see some recursive expression not having unique solutions.

   (a) Find at least two xs such that $xs = head\ xs : tail\ xs$.

   (b) Show that $xs := c + ones$ satisfy $xs = xs \curlyvee 1 + xs$. Since there is no restriction on the value of c, that implies that $xs = xs \curlyvee 1 + xs$ has infinite solutions for xs.

Solution: In fact, any stream satisfy $xs = head\ xs : tail\ xs$.

For the second problem, recall that

$$\begin{aligned} ones\ &= 0 : ones' \\ ones' &= 1 : ones' \curlyvee 1 + ones' \ . \end{aligned}$$

That is, $ones = 0 : 1 : ones' \curlyvee 1 + ones'$. We calculate:

$$
\begin{array}{ll}
& (c + ones) \curlyvee 1 + (c + ones) \\
= & \quad \{ \text{ definition of } ones \text{ and } (+) \ \} \\
& (c : c + ones') \curlyvee (1 + c : 1 + c + ones') \\
= & \quad \{ \text{ definition of } (\curlyvee) \text{, twice } \} \\
& c : 1 + c : c + ones' \curlyvee 1 + c + ones' \\
= & \quad \{ \text{ distributivity } \} \\
& c + (0 : 1 : ones' \curlyvee 1 + ones') \\
= & \quad \{ \ ones = 0 : 1 : ones' \curlyvee 1 + ones' \ \} \\
& c + ones \quad .
\end{array}
$$

4. Prove the distributivity law $n + (xs \curlyvee ys) = n + xs \curlyvee n + ys$, where n :: Int. Give a counter example showing that $zs + (xs \curlyvee ys) = zs + xs \curlyvee zs + ys$ does not hold in general.

Solution: Expanding $n + (xs \curlyvee ys)$:

$$
\begin{array}{ll}
& n + (xs \curlyvee ys) \\
= & \quad \{ \text{ definitions of } repeat \text{ and } (\curlyvee) \ \} \\
& (n : repeat\ n) + (head\ xs : ys \curlyvee tail\ xs) \\
= & \quad \{ \text{ definition of } (+) \ \} \\
& (n + head\ xs) : (n + (ys \curlyvee tail\ xs)) \quad .
\end{array}
$$

Expanding $n + xs \curlyvee n + ys$:

$$
\begin{array}{ll}
& n + xs \curlyvee n + ys \\
= & \quad \{ \text{ definitions of } repeat \ \} \\
& (n : repeat\ n) + xs \curlyvee n + ys \\
= & \quad \{ \text{ definition of } (+) \ \} \\
& ((n + head\ xs) : (n + tail\ xs)) \curlyvee n + ys \\
= & \quad \{ \text{ definition of } (\curlyvee) \ \} \\
& (n + head\ xs) : n + ys \curlyvee n + tail\ xs \quad .
\end{array}
$$

Both have the form $f\ n\ xs\ ys = (n + head\ xs) : f\ n\ ys\ (tail\ xs)$, which is an admissible equation. Therefore they must be equal.

Let $zs = nat$, we have

$$
\begin{array}{ll}
zs + (xs \curlyvee ys) & = 1 + x_0 : 2 + y_0 : 3 + x_1 : 4 + y_1 \ldots \\
zs + xs \curlyvee zs + ys & = 1 + x_0 : 1 + y_0 : 2 + x_1 : 2 + y_1 \ldots
\end{array}
$$

apparently not equal in general.

5. Prove that $(-1) \mathbin{\hat{}} nat = 1 \curlyvee (-1)$.

Solution: We start with expanding $(-1)\,\hat{}\,nat$ and try to find its recursive form:

$$\begin{aligned}
& (-1)\,\hat{}\,nat \\
= & \quad \{ \text{definition of } nat \} \\
& (-1)\,\hat{}\,(0 : 1 + nat) \\
= & \quad \{ \text{definition of } (\hat{}) \text{ (as a lifted binary operator)} \} \\
& (-1)\,\hat{}\,0 : (-1)\,\hat{}\,(1 + nat) \\
= & \quad \{ \text{arithmetics} \} \\
& 1 : (-1) \times (-1)\,\hat{}\,nat \quad .
\end{aligned}$$

Expanding $1 \curlyvee (-1)$:

$$\begin{aligned}
& 1 \curlyvee (-1) \\
= & \quad \{ \text{definition of } repeat \} \\
& (1 : repeat\ 1) \curlyvee (-1) \\
= & \quad \{ \text{definition of } (\curlyvee) \} \\
& 1 : (-1) \curlyvee 1 \\
= & \quad \{ \text{arithmetics} \} \\
& 1 : ((-1) \times 1) \curlyvee (-1 \times (-1)) \\
= & \quad \{ \text{distributivity} \} \\
& 1 : (-1) \times (1 \curlyvee (-1)) \quad .
\end{aligned}$$

Therefore $(-1)\,\hat{}\,nat = 1 \curlyvee (-1)$.

6. Given $nat = 0 : 1 + nat$ and $bin = 0 : 1 + 2 \times bin \curlyvee 2 + 2 \times bin$, prove that $nat = bin$.

Solution: It turns out to be easier to prove that nat matches the recursion pattern of bin, that is, $nat = 0 : 1 + 2 \times nat \curlyvee 2 + 2 \times nat$.

We have proved in the handouts that $nat = 2 \times nat \curlyvee 1 + 2 \times nat$. To be complete we recite the proof below:

$$\begin{aligned}
& 2 \times nat \curlyvee 1 + 2 \times nat \\
= & \quad \{ \text{definition of } nat \} \\
& 2 \times (0 : 1 + nat) \curlyvee 1 + 2 \times nat \\
= & \quad \{ \text{definition of } (\times) \} \\
& (0 : 2 \times (1 + nat)) \curlyvee 1 + 2 \times nat \\
= & \quad \{ \text{definition of } (\curlyvee) \} \\
& 0 : 1 + 2 \times nat \curlyvee 2 \times (1 + nat) \\
= & \quad \{ \text{arithmetics} \} \\
& 0 : 1 + (2 \times nat) \curlyvee 1 + (1 + 2 \times nat) \\
= & \quad \{ \text{distributivity} \} \\
& 0 : 1 + (2 \times nat \curlyvee 1 + 2 \times nat) \quad .
\end{aligned}$$

That is the recursion pattern of nat, therefore $2 \times nat \curlyvee 1 + 2 \times nat = nat$.

Therefore we have

$$
\begin{array}{ll}
& nat \\
= & \{ \text{definition of } nat \} \\
& 0 : 1 + nat \\
= & \{ \text{property above} \} \\
& 0 : 1 + (2 \times nat \curlyvee 1 + 2 \times nat) \\
= & \{ \text{distributivity, arithmetics} \} \\
& 0 : 1 + 2 \times nat \curlyvee 2 + 2 \times nat \quad .
\end{array}
$$

Therefore $nat = bin$.

7. Given a function h defined by:

$$
\begin{array}{ll}
h\ 0 & = k \\
h\ (1 + 2 \times n) & = f\ (h\ n) \\
h\ (2 + 2 \times n) & = g\ (h\ n) \quad ,
\end{array}
$$

prove that the sequence $map\ h\ bin = xs$ where

$$xs = k : map\ f\ xs \curlyvee map\ g\ xs \quad .$$

Solution: Note that the defintion says that

$$
\begin{array}{l}
h \cdot (1+) \cdot (2\times) = f \cdot h \quad , \\
h \cdot (2+) \cdot (2\times) = g \cdot h \quad .
\end{array}
$$

We calculate:

$$
\begin{array}{ll}
& map\ h\ bin = \\
= & \{ \text{definition of } bin \} \\
& map\ h\ (0 : 1 + 2 \times bin \curlyvee 2 + 2 \times bin) \\
= & \{ \text{definition of } map \text{ and } h \} \\
& k : map\ h\ (1 + 2 \times bin \curlyvee 2 + 2 \times bin) \\
= & \{ map \text{ distributes into } (\curlyvee) \} \\
& k : map\ h\ (1 + 2 \times bin) \curlyvee map\ h\ (2 + 2 \times bin) \\
= & \{ map \text{ fusion} \} \\
& k : map\ (h \cdot (1+) \cdot (2\times))\ bin \curlyvee map\ (h \cdot (2+) \cdot (2\times))\ bin \\
= & \{ \text{definition of } h \} \\
& k : map\ f\ (map\ h\ bin) \curlyvee map\ g\ (map\ h\ bin) \quad .
\end{array}
$$

which matches the recursion pattern of xs.

8. Prove that $nat \mathbin{/} 2 = nat \curlyvee nat$, where $(/)$ is integral division.

Solution: Expanding $nat \mathbin{/} 2$:

$$
\begin{aligned}
& nat \mathbin{/} 2 \\
= {} & \quad \{ \text{ definition of } nat \ \} \\
& (0 : 1 + nat) \mathbin{/} 2 \\
= {} & \quad \{ \text{ definition of } (/) \text{ (as a lifted binary operator) } \} \\
& (0 \mathbin{/} 2) : (1 + nat) \mathbin{/} 2 \\
= {} & \quad \{ \text{ definition of } nat \ \} \\
& 0 : (1 + (0 : 1 + nat)) \mathbin{/} 2 \\
= {} & \quad \{ \text{ definition of } (+) \ \} \\
& 0 : (1 : 2 + nat) \mathbin{/} 2 \\
= {} & \quad \{ \text{ definition of } (/)\text{, arithmetics } \} \\
& 0 : 0 : 1 + nat \mathbin{/} 2 \ \ .
\end{aligned}
$$

Expanding $nat \curlyvee nat$:

$$
\begin{aligned}
& nat \curlyvee nat \\
= {} & \quad \{ \text{ definition of } nat \ \} \\
& (0 : 1 + nat) \curlyvee nat \\
= {} & \quad \{ \text{ definition of } (\curlyvee) \ \} \\
& 0 : nat \curlyvee 1 + nat \\
= {} & \quad \{ \text{ definition of } (nat) \ \} \\
& 0 : (0 : 1 + nat) \curlyvee 1 + nat \\
= {} & \quad \{ \text{ definition of } (\curlyvee) \ \} \\
& 0 : 0 : 1 + nat \curlyvee 1 + nat \\
= {} & \quad \{ \ (1+) \text{ distributes into } (\curlyvee) \ \} \\
& 0 : 0 : 1 + (nat \curlyvee nat) \\
& \ \ .
\end{aligned}
$$

Both have form $\mathsf{X} = 0 : 0 : 1 + \mathsf{X}$, therefore $nat \mathbin{/} 2 = nat \curlyvee nat$.

9. Prove the abide law: $(fs \langle * \rangle xs) \curlyvee (gs \langle * \rangle ys) = (fs \curlyvee gs) \langle * \rangle (xs \curlyvee ys)$

Solution: Expanding $(fs \langle * \rangle xs) \curlyvee (gs \langle * \rangle ys)$:

$$
\begin{aligned}
& (fs \langle * \rangle xs) \curlyvee (gs \langle * \rangle ys) \\
= {} & \quad \{ \text{ definition of } (\langle * \rangle) \ \} \\
& (head\ fs\ (head\ xs) : tail\ fs \langle * \rangle tail\ xs) \curlyvee (gs \langle * \rangle ys) \\
= {} & \quad \{ \text{ definition of } (\curlyvee) \ \} \\
& head\ fs\ (head\ xs) : (gs \langle * \rangle ys) \curlyvee (tail\ fs \langle * \rangle tail\ xs) \ \ .
\end{aligned}
$$

Expanding $(fs \curlyvee gs) \langle * \rangle (xs \curlyvee ys)$:

$$
\begin{aligned}
& (fs \curlyvee gs) \langle * \rangle (xs \curlyvee ys) \\
= \quad & \{ \text{ definition of } (\curlyvee) \ \} \\
& (head\ fs : gs \curlyvee tail\ fs) \langle * \rangle (head\ xs : ys \curlyvee tail\ xs) \\
= \quad & \{ \text{ definition of } (\langle * \rangle) \ \} \\
& head\ fs\ (head\ xs) : (gs \curlyvee tail\ fs) \langle * \rangle (ys \curlyvee tail\ xs) \quad .
\end{aligned}
$$

Both have the form $f\ \ fs\ \ xs\ \ gs\ \ ys = head\ \ fs\ \ (head\ \ xs) : f\ \ gs\ \ ys\ \ (tail\ \ fs)\ \ (tail\ \ xs)$, therefore they are equal.