Functional Programming
Exercise 4: Program Calculation

Shin-Cheng Mu

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

1. Recall the standard definition of Fibonacci:

$$\begin{array}{l} \textbf{let}\ \mathit{fib}\ =\ \textbf{function} \\ \quad \mid\ 0 \to 0 \\ \quad \mid\ 1 \to 1 \\ \quad \mid\ \mathbf{1}{+}\ (\mathbf{1}{+}\ n) \to \mathit{fib}\ (\mathbf{1}{+}\ n) + \mathit{fib}\ n \end{array}$$

Let us try to derive a linear-time, tail-recursive algorithm computing *fib*.

   1. Given the definition $\mathit{ffib}\ n\ x\ y = \mathit{fib}\ n \times x + \mathit{fib}\ (n+1)\ \times y$. Express *fib* using *ffib*.
   2. Derive a linear-time version of *ffib*.

Solution: $\mathit{fib}\ n = \mathit{ffib}\ n\ 1\ 0$.

To construct *ffib*, we calculate:
Case 0:

$$\begin{array}{ll} & \mathit{ffib}\ 0\ x\ y \\ = & \{\ \text{definition of } \mathit{ffib}\ \} \\ & \mathit{fib}\ 0 \times x + \mathit{fib}\ 1 \times y \\ = & \{\ \text{definition of } \mathit{fib}\ \} \\ & 0 \times x + 1 \times y \\ = & \mathit{arithmetics} \\ & y \end{array}$$

Case $\mathbf{1}{+}\ n$:

$$\begin{array}{ll} & \mathit{ffib}\ (\mathbf{1}{+}\ n)\ x\ y \\ = & \{\ \text{definition of } \mathit{ffib}\ \} \\ & \mathit{fib}\ (\mathbf{1}{+}\ n) \times x + \mathit{fib}\ (\mathbf{1}{+}\ (\mathbf{1}{+}\ n)) \times y \\ = & \{\ \text{definition of } \mathit{fib}\ \} \\ & \mathit{fib}\ (\mathbf{1}{+}\ n) \times x + (\mathit{fib}\ (\mathbf{1}{+}\ n) + \mathit{fib}\ n) \times y \\ = & \{\ \text{arithmetics}\ \} \\ & \mathit{fib}\ (\mathbf{1}{+}\ n) \times (x + y) + \mathit{fib}\ n \times y \\ = & \{\ \text{definition of } \mathit{ffib}\ \} \\ & \mathit{ffib}\ n\ y\ (x + y) \end{array}$$

Therefore,

$$
\begin{aligned}
&\textbf{let } \mathit{ffib}\ n\ x\ y\ =\ \textbf{match } n \textbf{ with}\\
&\quad \mid\ 0 \to y\\
&\quad \mid\ \mathbf{1}\!+\ n \to \mathit{ffib}\ n\ y\ (x+y)
\end{aligned}
$$