1 Functional Reactive Programming

Consider the core calculus for Elm, presented in Lecture 24. Let \( i \) be a signal of integers (i.e., \( i \) has type \texttt{signal int}).

(a) Write a program that computes the factorial of the current value of \( i \). That is, the type of your program should be \texttt{signal int}, where the current value of the signal should be the factorial of the current value of \( i \). (Assume that you have whatever arithmetic operations you need.)

(b) Write a program that computes a signal that is the sum of all of the values of \( i \). That is, the type of your program should be \texttt{signal int}, where the current value of the signal should be the sum of all values that signal \( i \) took on. (Hint: use \texttt{foldp}. Assume that you have whatever arithmetic operations you need.)

(c) Write a program that computes a signal that is the sum of the current and the previous value of \( i \). Hint: you may assume that you have pairs.

(d) Write a program that attempts to use a signal of signals of integers. That is, write an expression that should have type \texttt{signal signal int}. Check to make sure that this expression is \textit{not} well typed.

(e) Show the first phase evaluation of the following program. Assume that \( i \) and \( j \) have type \texttt{signal int}.

\[
\begin{align*}
\text{let } \text{mul} &= \lambda a: \texttt{int}. \lambda b: \texttt{int}. \lambda c: \texttt{int}. a \times (b + c) \text{ in} \\
\text{let } \text{comb} &= \lambda x: \texttt{signal int}. \lambda y: \texttt{signal int}. \texttt{lift2 (mul) x y in}
\end{align*}
\]

\[
\begin{align*}
\text{let } t &= \text{comb } i \ j \text{ in} \\
\text{let } u &= \text{foldp (mul) 0 } t \text{ in} \\
\text{comb } i \ u
\end{align*}
\]

For the final term that is the result of the first phase evaluation of the program, draw a signal graph that shows the signals the program computes.

As a bonus, try the second phase evaluation of the program, assuming that the initial value of \( j \) is 1, and input signal \( i \) takes on the values 1, 2, 3, 4, \ldots.